

**Titre:** Un algorithme d'échantillonnage permettant d'améliorer la performance prédictive d'un modèle bayésien dans le cadre d'applications de tests adaptatifs  
**Title:**

**Auteur:** Alejandro Villarreal Morales  
**Author:**

**Date:** 2009

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Villarreal Morales, A. (2009). Un algorithme d'échantillonnage permettant d'améliorer la performance prédictive d'un modèle bayésien dans le cadre d'applications de tests adaptatifs [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/118/>  
**Citation:**

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/118/>  
**PolyPublie URL:**

**Directeurs de recherche:** Michel Desmarais, & Michel Gagnon  
**Advisors:**

**Programme:** Génie informatique  
**Program:**

UNIVERSITÉ DE MONTRÉAL

UN ALGORITHME D'ÉCHANTILLONNAGE PERMETTANT D'AMÉLIORER LA  
PERFORMANCE PRÉDICTIVE D'UN MODÈLE BAYÉSIEN DANS LE CADRE  
D'APPLICATIONS DE TESTS ADAPTATIFS

ALEJANDRO VILLARREAL MORALES  
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE INFORMATIQUE)  
AVRIL 2009

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

CE MÉMOIRE INTITULÉ :

UN ALGORITHME D'ÉCHANTILLONNAGE PERMETTANT D'AMÉLIORER LA  
PERFORMANCE PRÉDICTIVE D'UN MODÈLE BAYÉSIEN DANS LE CADRE  
D'APPLICATIONS DE TESTS ADAPTATIFS

présenté par : VILLARREAL MORALES Alejandro  
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées  
a été dûment accepté par le jury d'examen constitué de :

M. PIERRE Samuel, Ph.D., président

M. DESMARAIS Michel, Ph.D., membre et directeur de recherche

M. GAGNON Michel, Ph.D., membre et codirecteur de recherche

M. GALINIER Philippe, Doct., membre

*À ma femme Myriam, ma mère Carmen et mon grand-  
père Jesus.*

## **REMERCIEMENTS**

D'abord, je voudrais remercier Dieu de m'avoir donné la vie et les autres événements qui me sont arrivés.

Il y a plusieurs personnes que je voudrais remercier pour leur contribution à ce travail. Je voudrais remercier mon directeur de thèse, le Pr. Michel Desmarais et mon codirecteur, Pr. Michel Gagnon, qui m'ont accompagné toutes ces années. Merci pour tout le temps consacré à mon travail et par leurs conseils.

Je voudrais remercier ma femme Myriam pour son support. Je voudrais remercier spécialement ma mère qui m'a donné son amour, son attention et son aide durant toutes les années que j'ai pris pour finir. Je voudrais remercier mon grand-père pour son orientation dans ma vie et pour son support durant ces années. Je voudrais remercier ma tante Yola, mon oncle Javier et mes cousins Javier et Denise.

Je voudrais remercier aussi mes amis Peyman Meshkinfam, Xiaoming Pu, Belkacem Derfoul et Anka Stoykova pour leur soutien et leur aide.

Merci.

## RÉSUMÉ

Les systèmes CAT (abréviation anglaise de: Computing Adaptive Testing) sont utilisés depuis plusieurs décennies pour déterminer le niveau de connaissances des utilisateurs par le biais d'un questionnaire où l'on essaie de minimiser le nombre de questions posées. Plusieurs applications comme les systèmes de guidage et les tutoriels les utilisent afin de créer des profils utilisateurs.

Les systèmes CAT effectuent leur diagnostic sur la base d'un modèle de connaissances. Bien que certains modèles sont dérivés du travail d'un expert, comme les modèles à base de règles ou les modèles Bayésiens graphiques, où la structure et les probabilités conditionnelles sont définies par un expert du domaine, les modèles les plus prometteurs sont basés sur une approche statistique où les paramètres du modèle sont induits à partir de données. (Desmarais et Gagnon, 2006; Millan et Perez-de-la-cruz, 2002; Vomlel, 2004)

La construction d'une banque de données pour créer le modèle de connaissances est une tâche parfois laborieuse, puisqu'il faut créer les questionnaires et soumettre ces questionnaires à un ensemble de personnes afin de recueillir ces données. Le problème se pose plus particulièrement lorsqu'il faut créer un modèle détaillé comportant plusieurs connaissances et impliquant ainsi de nombreuses questions. Il est pratiquement impossible d'amasser des données complètes pour un apprenant lorsque le modèle comporte des centaines, voir des milliers de questions, car la réponse à un tel questionnaire nécessiterait trop de temps pour un même répondant. Il faut donc administrer des sous-ensembles de questions à un plus grand nombre de répondants.

Dans ce mémoire, nous explorons la création d'une banque de données en utilisant des données partielles. Nous visons à entraîner un modèle de connaissances à partir de données provenant de questionnaires partiels, comportant un sous-ensemble de questions. L'objectif spécifique du mémoire est de vérifier si la performance prédictive du modèle d'inférence et d'évaluation des connaissances peut être améliorée avec un

schème d'échantillonnage des données adapté, c'est-à-dire en déterminant la composition des questionnaires utilisés pour le recueil de données.

L'analyse permettant de déterminer quelles questions seront prises en compte dans la composition des questionnaires sera faite dans le cadre de l'approche de modélisation de la connaissance POKS (abréviation en anglais de: Partial Order Knowledge Structure), qui est un modèle bayésien graphique.

La question étudiée dans ce mémoire est donc de savoir si la composition des questionnaires d'entraînement peut être optimisée pour obtenir les modèles les plus performants possible. Par exemple, est-il possible qu'en augmentant la fréquence relative de certains items dans ces questionnaires, nous obtenions des modèles dont le diagnostic est meilleur ?

Nous étudions ces questions avec une approche empirique et expérimentale. Nous utilisons quatre types d'échantillons: complets, aléatoires, à incertitude maximale et à incertitude minimale. Les deux premiers types servent aux fins de comparaison alors que les deux derniers représentent respectivement des données partielles comportant dans un cas un plus grand nombre de questions dont le taux de réussite est près de 50% (incertitude maximale) et dans un autre cas des questions dont le taux de réussite se rapproche de 0% ou 100% (incertitude minimale). L'expérimentation consiste à simuler un processus de test adaptatif avec un répondant et à comparer les inférences du modèle avec les réponses réelles du répondant. Ce processus consiste à choisir la question qui apportera le plus d'information à la lumière des réponses précédentes. La performance est mesurée après chaque question posée.

Les résultats obtenus démontrent que l'échantillonnage à incertitude minimale donne le meilleur résultat et se rapproche de la performance de l'échantillonnage complet. Ces résultats se sont avérés contraires aux attentes, car nous présumions initialement que l'échantillonnage à incertitude maximale fournirait les meilleurs résultats. Une explication plausible est les questions à incertitude minimale comportent un plus grand bruit d'échantillonnage que les autres, et que ce bruit induit des erreurs d'inférences

importantes. En augmentant leur taux d'échantillonnage, ce bruit est alors réduit et l'exactitude des inférences augmente.



## ABSTRACT

Bayesian graphical models are commonly used to build student models from data. A number of standard algorithms are available to train Bayesian models from student skills assessment data. These models can assess student knowledge and skills from a few observations. They are useful for Computer Adaptive Testing (CAT), for example, where the test items can be administered in order to maximize the information they will provide. In practice, such data often contain missing values and, under some circumstances, missing values far outnumber observed values. However, when collecting data from test results, one can often choose which values will be present or missing by a consequent test design. We study how to optimize the choice of test items for collecting the data that will be used for training a Bayesian CAT model, such as to maximize the predictive performance of the model. We explore the use of a simple heuristic for test item choice based on the level of uncertainty. The uncertainty of an item is derived from its initial probability of success and, thus, from its difficulty.

The results show that this choice does affect model performance and that the heuristic can lead to better performance. Although the study's results are more exploratory than conclusive, they suggest interesting research avenues.

## TABLE DES MATIÈRES

DÉDICACE .....	iii
REMERCIEMENTS.....	iv
RÉSUMÉ .....	v
ABSTRACT.....	viii
TABLE DES MATIÈRES .....	ix
LISTE DES TABLEAUX.....	xi
LISTE DES FIGURES .....	xii
LISTE DES NOTATIONS ET DES SYMBOLES .....	xiii
CHAPITRE 1 INTRODUCTION .....	1
1.1 Problématique .....	1
1.2 Objectifs de la recherche.....	2
1.3 Organisation.....	3
CHAPITRE 2 APPLICATIONS DES TESTS ADAPTATIFS POUR L'APPRENTISSAGE PERSONNALISÉ .....	4
2.1 Personnalisation et modèle de connaissance.....	4
2.2 Les tests adaptatifs pour le guidage et l'évaluation .....	5
2.2.1 ALEKS.....	5
2.2.2 Brainbench .....	7
2.3 Les tutoriels.....	9
2.3.1 Andes .....	10
2.3.2 Netcoach .....	12
2.3.3 Leactivemath.....	16
2.4 Résumé.....	20

CHAPITRE 3	MODÈLES DE LA CONNAISSANCE .....	22
3.1	Modèle de la Théorie des Réponses aux Items .....	22
3.2	Les espaces de connaissances .....	25
CHAPITRE 4	LE MODÈLE POKS .....	30
4.1	Définitions .....	30
4.2	Les liens .....	31
4.3	Actualisation des probabilités .....	33
4.3.1	Sélection des nœuds évidence .....	34
4.3.2	Probabilité postérieure .....	34
4.4	Résumé .....	36
CHAPITRE 5	MÉTHODOLOGIE .....	37
5.1	Retour sur la problématique et les objectifs .....	37
5.2	Conditions expérimentales .....	38
5.3	Méthodes d'échantillonnage .....	39
5.3.1	Algorithme d'échantillonnage .....	40
5.4	Simulations et mesure de la performance d'un modèle POKS .....	43
5.4.1	Algorithme de simulation .....	44
5.4.2	Algorithme d'engine .....	46
5.4.3	Algorithme d'inference .....	47
5.5	Banque de données .....	48
CHAPITRE 6	RÉSULTATS .....	50
CONCLUSION	.....	57
7.1	Synthèse des travaux .....	57
7.2	Limitations des travaux .....	58
7.3	Recherches futures .....	59
RÉFÉRENCES	.....	61

**LISTE DES TABLEAUX**

Tableau 1. Tableau de contingence des possibilités de combinaisons des items $X_a$ et $X_b$ .....	32
--	----

## LISTE DES FIGURES

Figure 1. Graphique circulaire qui montre les résultats de ALEKS .....	6
Figure 2. Exemple des messages d'erreur d'ALEKS .....	7
Figure 3. Comment Brainbench la personne adéquate pour un poste donné.....	8
Figure 4. Feuille de recommandation de Brainbench .....	9
Figure 5. Exemple d'un problème de physique et sa solution dans Andes .....	11
Figure 6. Exemple d'un environnement adaptatif dans NetCoach.....	13
Figure 7. Interface en ligne des tuteurs dans NetCoach.....	15
Figure 8. Page principale de LeActiveMath .....	17
Figure 9. Outil interactif pour travailler avec les équations linéaires .....	18
Figure 10. Exemple graphique des 8 composants de l'architecture de LeActiveMath .....	20
Figure 11. La structure du modèle TRI.....	23
Figure 12. Exemple de la graphique d'une courbe ICC .....	24
Figure 13. Espace de connaissances. Chaque sous-ensemble représente un état possible de la connaissance d'un individu. ....	26
Figure 14. Structure de connaissances représentant l'ordre d'apprentissage équivalent à la Figure 13. ....	27
Figure 15. Exemple des liaisons des items dans un espace de connaissances.....	28
Figure 16. Distribution de la probabilité d'échantillonnage des items en fonction de leur difficulté (rang en percentile).....	40
Figure 17. Graphique de performance pour la banque de données Mathématiques.....	51
Figure 18. Graphique de performance pour la banque de données Unix.....	52
Figure 19. Graphique de performance pour la banque de données Français .....	53
Figure 20. Graphique de performance pour la banque de données Arithmétique .....	54

**LISTE DES NOTATIONS ET DES SYMBOLES**

$\Psi$	Ensemble des connaissances d'un domaine
$S_j$	État de connaissance d'un sujet $j$
$Bp(k, n, p)$	Fonction binomiale ( $k$ : succès, $n$ : essais, $p$ : probabilité de succès)
$ \mathbf{X} $	Cardinalité de l'ensemble $\mathbf{X}$
$\mu$	Moyenne (d'une population)
$\sigma$	Écart type (d'une population)
$\varphi(x)$	Probabilité de $x$ d'une distribution normale
$\theta$	Habileté (dans le cadre de la TRI)

## CHAPITRE 1 INTRODUCTION

### 1.1 Problématique

Les tests adaptatifs permettent de minimiser le nombre de questions posées à un individu pour arriver à un diagnostic sur ses connaissances acquises. Depuis plusieurs années ils sont couramment utilisés pour les examens d'entrée aux universités et autres institutions.

Les modèles utilisateur sont utilisés dans les environnements d'apprentissage intelligent pour créer des tests adaptatifs qui sont eux-mêmes utilisés pour faire des diagnostics de l'état de connaissance des utilisateurs (Garrido, 2006). Quelques exemples de tests adaptatifs bien connus sont celui de TOEFL, les tests de certification de Microsoft ou encore les tests de certification de Cisco.

Les tests adaptatifs sont une façon de mesurer les connaissances des utilisateurs. À partir du succès ou de l'échec de la réponse à une question posée à un utilisateur, le système inférera sa maîtrise d'autres compétences et la prochaine question posée sera celle qui apporte le plus d'information. On réduit ainsi le nombre de questions posées et on arrive à un diagnostic rapidement.

Pour construire le modèle de connaissances derrière les inférences et le diagnostic, il faut recueillir une grande quantité de données. D'une part, il faut un nombre minimal de réponses à chaque question pour calibrer les modèles. D'autre part, il faut aussi un grand nombre de questions pour différentes raisons : (1) pour couvrir adéquatement le domaine de connaissances à tous les niveaux d'expertise, comme par exemple la maîtrise écrite d'une langue; et (2) pour varier les questions. En effet, plus une question est fréquemment posée, plus les chances que la réponse soit connue augmentent. De plus, pour des applications comme un tutoriel ou un guide d'étude, il est courant de devoir fréquemment questionner l'apprenant et il importe donc d'avoir une grande banque de questions concernant les mêmes compétences. En pratique, il est donc courant de construire une base de questions de plusieurs centaines de questions. Par exemple, c'est le cas du test

de mathématique que l'École Polytechnique de Montréal utilise pour les nouveaux inscrits : chaque année, l'École offre aux nouveaux arrivants de passer un test de 60 questions en mathématique pour leur fournir un avis sur leur niveau de préparation aux cours de mathématique de première année. Or, la banque de questions comporte près de 1000 items (questions). Même si le test n'est pas encore adaptatif, ce grand nombre d'items permet d'éviter le problème de sur-exposition des items en créant des tests différents.

Le problème pour créer de grandes banques de questions est qu'on ne peut obtenir les données de réponses à toutes les questions pour un même répondant. En effet, on ne peut demander à un seul répondant de répondre à plusieurs centaines de questions. Il faut donc recourir à des données partielles pour entraîner un modèle de connaissances. C'est-à-dire que pour un test de 1000 questions, par exemple, il faut créer des tests de 50 à 100 questions et les passer à plusieurs répondants pour couvrir l'ensemble des questions au moins une fois chacune.

Le choix des questions par répondant peut être fait en assurant un nombre plus ou moins identique de répondants par question, soit par échantillonnage totalement aléatoire, soit par un choix systématique des questions. Mais ce ne sont pas nécessairement des approches optimales. Il est possible que certaines questions nécessitent un plus grand nombre de répondants aux dépens d'autres questions. Nous abordons dans ce mémoire la question à savoir s'il est possible, variant le nombre de répondants par question, de construire un modèle de l'apprenant plus performant pour le diagnostic des compétences avec un test adaptatif qu'une approche qui assure un nombre égal de répondants par question.

## **1.2 Objectifs de la recherche**

Les tests adaptatifs se basent sur un modèle pour diagnostiquer les compétences et réduire le nombre de questions afin d'établir le profil de connaissance d'un apprenant. Ce modèle est construit à partir de données de réponses aux items d'un test, mais ces



données sont généralement partielles, dans le sens où un même répondant ne répond qu'à un sous-ensemble des questions d'un test.

L'objectif de cette étude est de déterminer un algorithme de choix des items par répondant (c.-à-d. un algorithme d'échantillonnage) qui permettra au modèle construit à partir de ces données d'être plus performant pour diagnostiquer les compétences qu'un modèle construit à partir d'un choix aléatoire. Nous verrons dans le chapitre sur la méthodologie que deux méthodes sont proposées dans le but d'améliorer la performance du diagnostic. Une première consiste à choisir des questions dont le taux de succès est moyen, et une seconde à choisir des questions faciles et difficiles en priorité. Les premières correspondent à des questions à forte entropie et les secondes à faible entropie.

### **1.3 Organisation**

Dans le prochain chapitre, nous révisons les applications des tests adaptatifs et les autres applications qui utilisent les modèles de connaissances pour offrir une forme personnalisée d'apprentissage.

Nous allons réviser certains de ces modèles. La TRI (Théorie des Réponses aux Items) est l'approche la plus répandue pour la modélisation des compétences dans le domaine des tests adaptatifs. Nous la présentons au chapitre 3, de même que la théorie des espaces de connaissances sur laquelle le modèle utilisé dans l'étude est basé.

Le chapitre 4 est consacré à un type spécifique de modèle de la connaissance, l'approche POKS. L'étude est effectuée avec ce modèle et il est décrit en détail dans ce chapitre.

La méthodologie de l'étude suit au chapitre 5 et nous compléterons avec la présentation, l'analyse et la discussion des résultats au chapitre 6.

Avant d'aborder les modèles de connaissances, le chapitre 2 se consacre à la description des applications de guidage et de tutoriel qui utilisent les tests adaptatifs et qui sont utilisés dans l'apprentissage personnalisé.

## **CHAPITRE 2 APPLICATIONS DES TESTS ADAPTATIFS POUR L'APPRENTISSAGE PERSONNALISÉ**

Dans ce chapitre, nous allons revoir les applications qui font l'usage de modèles de connaissances, notamment les applications utilisées pour l'apprentissage personnalisé. Nous allons porter une attention particulière aux guides d'études et aux tutoriels. Toutes ces applications peuvent bénéficier d'un modèle de la connaissance et de moyens de diagnostic rapide de la connaissance comme les tests adaptatifs.

### **2.1 Personnalisation et modèle de connaissance**

Les modèles de connaissance servent avant tout à personnaliser le tutorat dans un système d'apprentissage assisté par ordinateur.

En effet, la personnalisation du processus de tutorat est un objectif très important. Le phénomène 2 sigma en atteste. Il stipule qu'en utilisant un tutoriel un à un, donc personnalisé, on obtient une performance supérieure de 2 sigma sur la moyenne des étudiantes qui ont appris la même information avec un moyen conventionnel d'apprentissage non personnalisé (Bloom, 1984). C'est-à-dire qu'en utilisant les techniques d'apprentissage personnalisés avec des élèves, nous obtenons une amélioration de 2 sigma par rapport à un groupe qui a reçu une formation en groupe.

Selon les articles de (Heller, Steiner, Hockemeyer et Albert, 2006 ; Martinez, 2002 ; Xiaolin, 2002) l'apprentissage personnalisé est décrit comme l'adaptation du matériel didactique présenté à l'apprenant en fonction des besoins, des préférences, des intérêts et des aspirations des étudiants. En utilisant cette information, un système peut créer des stratégies d'apprentissage personnalisées à chaque étudiant. Évidemment, la personnalisation doit reposer sur un diagnostic précis des connaissances et des intérêts.

## **2.2 Les tests adaptatifs pour le guidage et l'évaluation**

Le guidage consiste à proposer à l'apprenant le matériel didactique à consulter qui est le plus pertinent pour cheminer progressivement vers un objectif d'apprentissage. Il existe plusieurs applications de ce type.

Nous révisons quelques exemples d'applications de guidage dans ce qui suit.

### **2.2.1 ALEKS**

ALEKS (Aleks Corporation, 2008) est une application qui utilise un modèle de connaissance sophistiqué pour guider l'apprenant à travers des exercices. L'apprenant passe en premier un court test adaptatif d'une vingtaine de minutes pour établir un diagnostic de sa connaissance, puis il peut utiliser ALEKS pour pratiquer les notions à apprendre dans un contexte personnalisé et adapté au diagnostic effectué.

ALEKS a été développé à l'université de New York et à l'université de la Californie par une équipe d'ingénieurs de logiciels, des mathématiciens et des scientifiques cognitifs. Le cœur d'ALEKS est un engin d'inférence basé sur la théorie des espaces de connaissance (Falmagne et Doignon, 2003). Une fois le test adaptatif complété, ALEKS affiche les résultats à l'aide d'un graphique circulaire (Figure 1). Ce graphique décrit la connaissance actuelle de l'élève et il fait des recommandations pour des sujets d'études pertinentes à l'élève. Si l'étudiant clique sur le graphique, le mode d'apprentissage commence. ALEKS fera la suggestion de quelques matières à apprendre.

Au moment où l'élève fait son choix, ALEKS lui offre des problèmes de pratique pour apprendre la matière. Si l'étudiant ne comprend pas quelque chose, il peut toujours accéder à une explication complète de la matière. Quand l'étudiant donne des réponses correctes aux problèmes que ALEKS lui pose, ALEKS considère que l'étudiant a appris la matière et l'élève peut choisir une autre matière à apprendre.

Quand l'étudiant donne des réponses incorrectes, ALEKS montre un message d'erreur (Figure 2) et lui donne des suggestions pour faire la correction des erreurs

ALEKS pourrait aussi proposer à l'étudiant d'abandonner temporairement le problème pour lire plus d'information et ensuite réessayer de résoudre le problème.

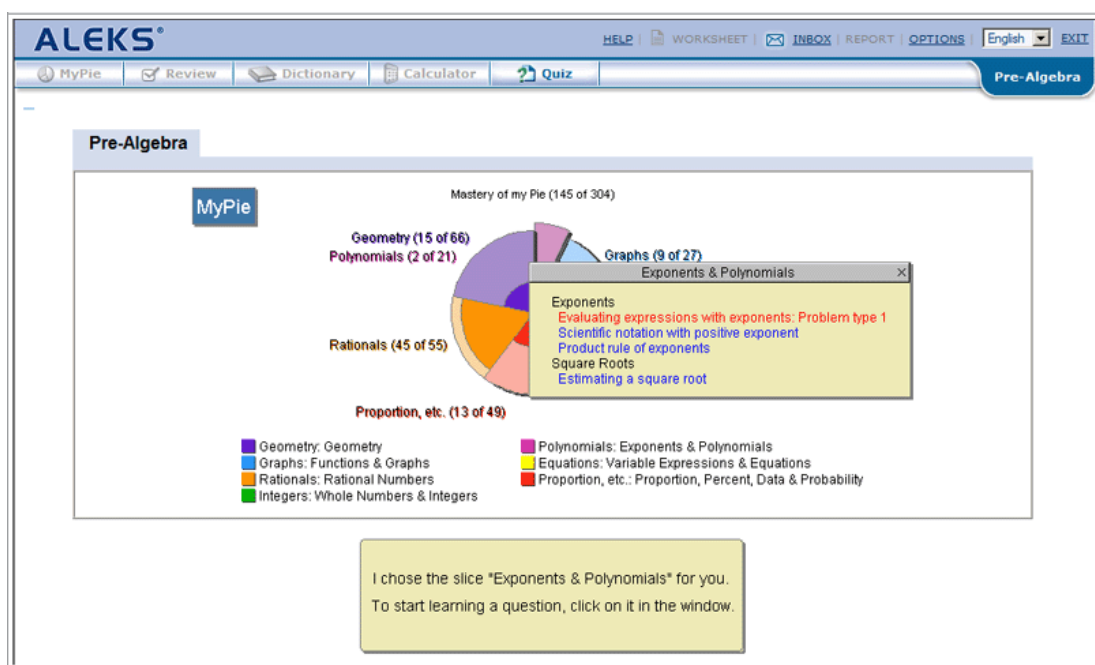


Figure 1 Graphique circulaire qui montre les résultats de ALEKS (www.aleks.com)

ALEKS offre l'avantage de fournir une instruction personnalisée, un à un, disponible en tout temps à un prix bien en deçà d'un tuteur humain (Aleks Corporation, 2008). Le guidage offert par ALEKS touche maintenant à plusieurs domaines comme la comptabilité, la chimie, et les mathématiques du niveau primaire au niveau universitaire. L'application peut être utilisée en mode individuel ou dans le cadre d'un programme d'étude.

Nous verrons plus loin qu'ALEKS utilise un modèle de diagnostic des compétences basé sur les espaces de connaissances qui a plusieurs points en commun avec le modèle utilisé pour cette étude, POKS.

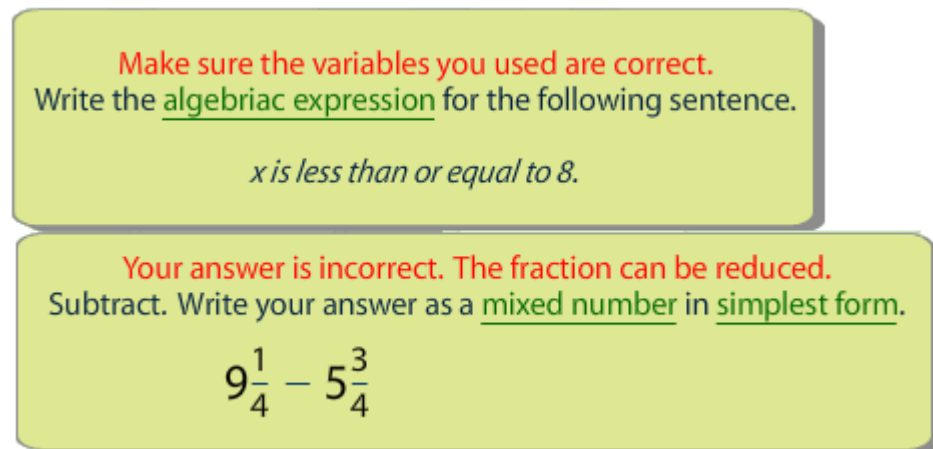


Figure 2 Exemple des messages d'erreur d'ALEKS ([www.aleks.com](http://www.aleks.com))

### 2.2.2 Brainbench

Brainbench ([www.brainbench.com](http://www.brainbench.com)) est un autre système qui utilise les tests adaptatifs pour le diagnostic des connaissances dans le domaine de la gestion des ressources humaines. Il donne des outils aux entreprises pour évaluer les personnes qui postulent pour un poste (Figure 3). L'outil identifie les connaissances, les habilités et les capacités de chaque individu. Le but est de trouver la personne la plus appropriée pour un poste donné.



Figure 3 Comment Brainbench identifie la personne adéquate pour un poste donné  
([www.brainbench.com/xml/bb/business/products.xml](http://www.brainbench.com/xml/bb/business/products.xml))

La mission de Brainbench est de donner des produits faciles à gérer qui peuvent faire des prédictions de succès dans le travail. Nous pouvons voir une feuille de recommandation dans la Figure 4.

Brainbench se démarque par le grand nombre de tests et une grande base de clients. Ils affirment avoir une banque de 600 tests et avoir administré jusqu'à 6 millions de tests à des individus et une base de 5000 clients entreprises. Il s'agit donc d'une des applications les plus importantes des tests adaptatifs.

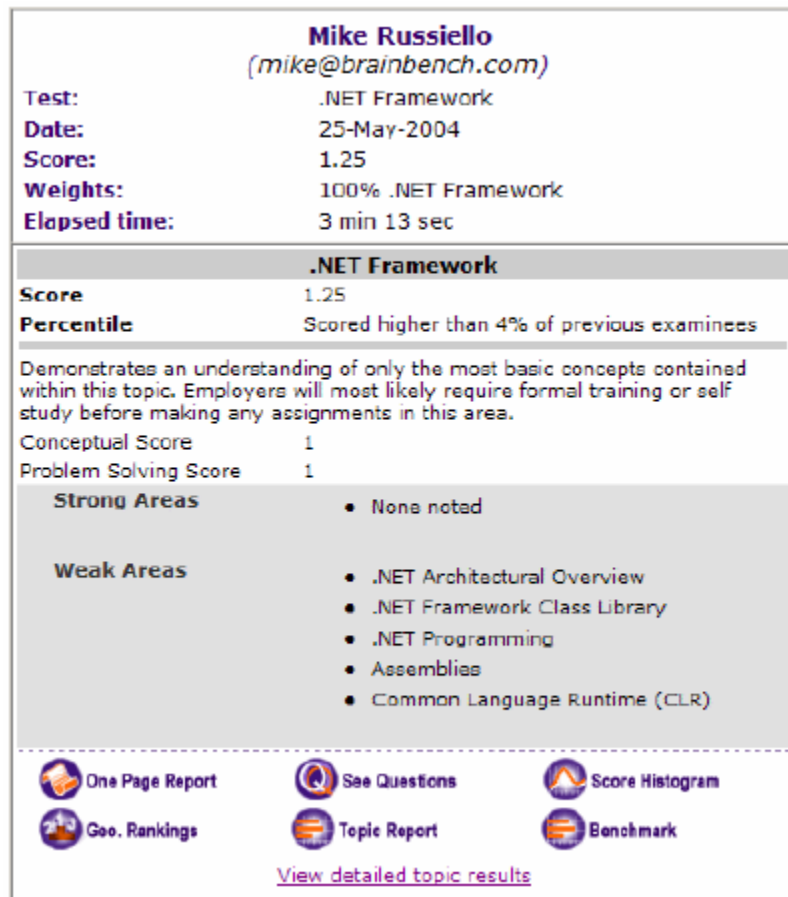


Figure 4 Feuille de recommandation de Brainbench  
 (www.brainbench.com/xml/bb/business/products.xml)

### 2.3 Les tutoriels

Les tutoriels intelligents sont une autre famille d'applications qui utilisent un modèle de la connaissance pour diagnostiquer les compétences d'un individu dans le but de personnaliser l'apprentissage. Les modèles sous-jacents peuvent en fait être les mêmes que ceux des tests adaptatifs, à la différence qu'ils n'ont pas un mode « test » en tant que tel. Ils font le diagnostic de la connaissance de l'apprenant directement à travers le processus d'apprentissage en observant ce que l'apprenant réussit ou échoue, et personnaliser le contenu en conséquence.

Un tutoriel est un logiciel avec l'objectif de donner une formation personnalisée à un utilisateur dans un domaine de connaissances spécifique. Le tutoriel doit faire l'inférence des connaissances de l'utilisateur portant sur le domaine d'intérêt, et ce, à partir de l'analyse de la performance de l'utilisateur. Nous pouvons mesurer la performance d'un utilisateur en posant des questions dans un test et à travers des exercices présentés. Le tutoriel peut donner deux types de conseils : les conseils actifs et les conseils passifs.

Les conseils actifs sont donnés directement à travers une intervention dans les actions de l'utilisateur. Les conseils passifs sont donnés à travers des réponses aux questions posées par l'utilisateur. De cette façon, l'élève apprendra une nouvelle connaissance et le tutoriel donnera une meilleure suggestion. Dans les deux méthodes, le tutoriel essayera de donner un conseil en rapport avec le niveau de connaissances qu'il croit que l'utilisateur maîtrise.

Quelques exemples de tuteurs sont décrits ici.

### **2.3.1 Andes**

Le didacticiel Andes est un didacticiel intelligent pour l'apprentissage de la physique, développé à l'University of Pittsburgh. Les étudiants lisent le problème, dessinent des vecteurs et mettent les coordonnées des axes, font la définition des variables et écrivent des équations. Le système affiche le résultat et donne à l'étudiant une rétroaction immédiate. Si l'information écrite par l'étudiant est correcte, la couleur des



mots sera verte et rouge dans le cas contraire.

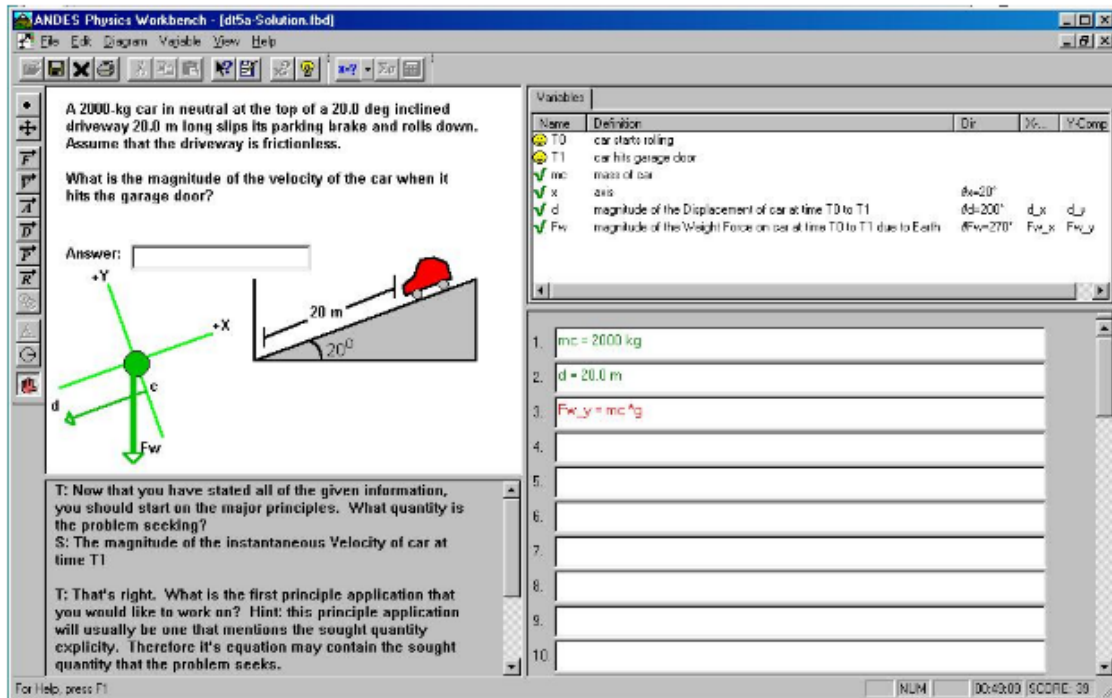


Figure 5 Exemple d'un problème de physique et sa solution dans Andes  
([www.andestutor.org/Pages/Reasons.html](http://www.andestutor.org/Pages/Reasons.html))

Andes donne une réponse à certains types d'erreurs produites par les utilisateurs.

Il utilise trois types d'interventions :

- 1.- Quand une erreur n'est pas reconnue par l'utilisateur, la couleur des mots passe au rouge. Par exemple, si une entrée n'a pas été décrite, si un bouton n'a pas été appuyé, etc.
- 2.- L'étudiant peut demander de l'aide à tout moment.
- 3.- Si l'étudiant n'est pas certain de ce qu'il doit faire après une étape, il peut demander un indice de la prochaine étape à suivre. Andes fournit alors une explication et des suggestions sur les actions possibles.

Chaque fois que l'étudiant résout un problème, ANDES montre les points gagnés par l'étudiant. ANDES essaye d'encourager les habitudes des étudiants pour donner de

bonnes solutions aux problèmes en base à l'attribution des points pour écrire quelques réponses de façon explicite. Les étudiants auront des points aussi grâce à l'écriture des équations basées sur des principes fondamentaux qui n'ont pas des valeurs données.

L'étudiant peut travailler avec ANDES en ligne et hors ligne. Si les étudiants travaillent hors ligne, ils peuvent imprimer le devoir. Si les étudiants travaillent avec ANDES en ligne, ils peuvent envoyer les réponses par l'internet. Le diagnostic des compétences de chaque étudiant est envoyé directement au professeur.

### **2.3.2 Netcoach**

NetCoach (Weber, Kuhl et Weibelzahl, 2002) est un système d'écriture qui permet de créer des cours individuels et adaptatifs, sans aucune connaissance en programmation. Les cours développés par NetCoach ont quatre caractéristiques en commun :

#### **1.- Adaptatifs**

NetCoach peut adapter la connaissance, les buts et les préférences des utilisateurs. Cette information peut être utilisée pour adapter la présentation du contenu de l'information ou pour aider dans la navigation. NetCoach utilise deux techniques de navigation adaptative :

a.- Séquence de curriculum : la séquence de curriculum aide les étudiants à trouver les meilleurs chemins à travers l'information.

b.- Annotation adaptative des liens : l'annotation adaptative des liens aide l'étudiant à s'orienter dans l'hyperespace en changeant l'apparence des liens.

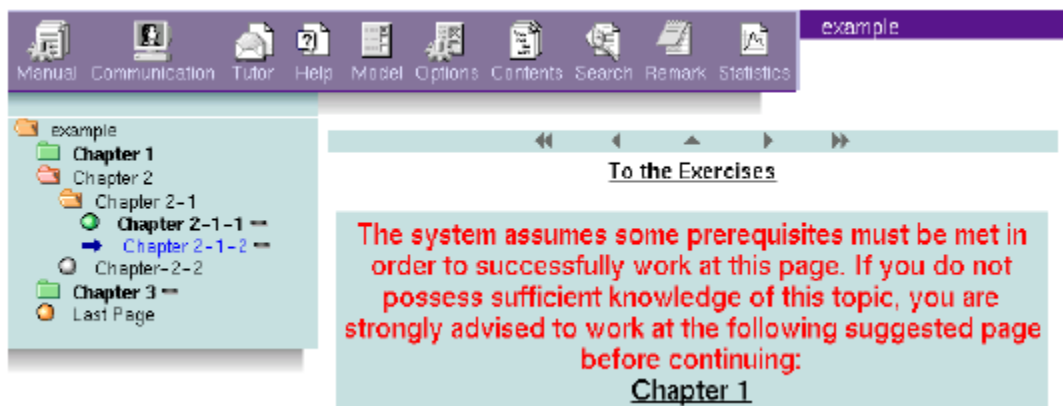


Figure 6 Exemple d'un environnement adaptatif dans NetCoach (Weber, Kuhl et Weibelzahl, 2002)

## 2.- Interactifs.

Un utilisateur peut apprendre dans un système s'il regarde l'information dans l'écran mais l'apprentissage est plus efficace si le système permet à l'utilisateur d'avoir un échange interactif. Dans NetCoach, les exercices, les tests et les animations offrent des moyens d'interactivité avec l'utilisateur. Les animations font les simulations de possibles scénarios de test. Les animations donnent la possibilité d'interagir avec l'utilisateur par le biais de la souris. Dans les animations, il y a des événements qui répondent au choix de l'utilisateur.

## 3.- Adaptables.

Au point 1, nous avons mentionné que le système peut s'adapter au profil de l'utilisateur. Les cours adaptables sont ceux dans lesquels l'utilisateur fait l'adaptation manuelle à travers du choix des options. Par exemple : la présentation des messages d'erreur et la présentation des recommandations.

#### 4.- Communicatifs.

La communication dans le système peut être synchrone ou asynchrone. Un exemple de communication synchrone est le programme de clavardage (chat) que l'élève peut utiliser avec son professeur. Un exemple de communication asynchrone est le courrier électronique que l'utilisateur peut envoyer à son professeur, contenant des exercices, des questions, etc.

Avec NetCoach, les concepts sont la base de connaissance d'un cours. Les concepts sont des représentations internes de pages qui seront montrées à l'étudiant. Pour construire la base de connaissance, l'auteur peut construire plusieurs relations de contenu spécifiques pour chaque concept.

Il y a deux types de relations entre concepts : pré-conditions et inférences. Le système fera le guidage des élèves à travers des pages des pré-conditions avant de faire la suggestion d'un concept.

##### **2.3.2.1 Les tuteurs dans les cours de Netcoach**

Les cours créés avec NetCoach sont animés par un tuteur qui aide le réalisateur d'un cours à créer l'aide, analyser l'information de l'élève, faire l'édition de listes de discussion, envoyer des messages aux utilisateurs, administrer les comptes des utilisateurs et administrer les groupes d'utilisateurs.

Les tuteurs ont leurs propres droits d'accès. Ils peuvent faire l'inspection des cours et des utilisateurs dans un cours.

Les réalisateurs d'un cours peuvent contacter les utilisateurs avec un écran d'aide dans le tuteur. Les utilisateurs peuvent poser des questions ou donner des commentaires dans le cours.

The screenshot displays the NetCoach tutor interface. At the top, there are buttons for 'show user statistics', 'load old user', and 'remove user', each with a right-pointing arrow. Below these is a text box showing 'Total of 4 users' and '3 active users in the last 60 minutes'. A 'show' button is next to it. To the right, a list of users is shown: 'joe', 'john', and 'student'. Further right, under the heading 'example', there are three columns: 'Users and User Groups', 'Discussion List', and 'User Infos', each with an 'edit' button. Below this, a table titled 'Active users in example (4)' shows user activity. At the bottom, a detailed view for user 'john' is shown, including 'Visited pages: 8', 'Learning time: 4 min, 36 sec', and 'Current concept: Chapter-2-1-2'. This is followed by a table of tasks and exercises, and a table of learning state points.

**example**

Users and User Groups    Discussion List    User Infos

edit    edit    edit

**Active users in example (4)**

Last Change	Load Time	User Name	Note	Act Item
14:01:22 June 20, 2001	13:54:50 June 20, 2001	<a href="#">john</a>	0	<a href="#">Chapter-2-1-2</a>
13:54:25 June 20, 2001	13:54:14 June 20, 2001	<a href="#">joe</a>	0	<a href="#">Chapter-3-1</a>
13:51:59 June 20, 2001	13:51:42 June 20, 2001	<a href="#">student</a>	0	<a href="#">Chapter-2-1</a>

**Name: john**  
Visited pages: 8  
Learning time: 4 min, 36 sec  
Current concept: Chapter-2-1-2

Tasks worked on	Current Page: Chapter 2-1-2	Chapter: Chapter 2-1	Total Course
Exercises	1.0 of 1.0 100.0% correct	2.0 of 2.0 100.0% correct	4.0 of 4.0 100.0% correct
Tests	-	-	-
Total	1.0 of 1.0 100.0% correct	2.0 of 2.0 100.0% correct	4.0 of 4.0 100.0% correct
Learning State (Points)	1.0 of 1.0 100.0% learned	2.0 of 2.0 100.0% learned	4.0 of 5.0 80.0% learned

User: john

Course	Concept	CritV	Value	Solv	Infer	Hits	Errs	NCorr	NFalse	U-No	ConceptName
2	2	1.0	1.0	1	0	1	0	1	0	0	Chapter-1-1
2	3	1.0	1.0	1	0	1	0	1	0	0	Chapter-1-2
2	6	1.0	1.0	1	0	1	0	1	0	0	Chapter-2-1-1
2	7	1.0	1.0	1	0	1	0	1	0	0	Chapter-2-1-2

Figure 7 Interface en ligne des tuteurs dans NetCoach (Weber, Kuhl et Weibelzahl, 2002)

Le texte écrit dans l'écran du tuteur est envoyé par le serveur, en utilisant le courrier électronique. Les tuteurs peuvent répondre à travers du courrier électronique ou les tuteurs peuvent envoyer de messages qui seront présentés aux utilisateurs quand ils ouvriront la prochaine page de leur cours.

Les tuteurs peuvent observer les utilisateurs dans les cours. Les tuteurs ont une liste de tous les utilisateurs actifs. Ils peuvent analyser l'état actuel d'apprentissage des utilisateurs. C'est-à-dire que les tuteurs peuvent analyser :

- Combien de temps ont travaillé les utilisateurs dans un cours ?

- Quels messages ont envoyés les utilisateurs ?
- Avec quels concepts ont travaillé les utilisateurs ?
- Combien d'erreurs ils ont fait ?

### **2.3.2.2 Le modèle utilisateur dans le Netcoach**

NetCoach utilise un modèle utilisateur multicouches. Les données individuelles de chaque élève sont enregistrées en relation directe avec les concepts de la base de données des cours. La première couche montre les pages que l'élève a déjà visitées. La deuxième couche porte sur les données des exercices et les items de test reliés à un concept avec lequel l'élève a travaillé sous un critère défini. La troisième couche montre si un concept peut être inféré à partir des concepts déjà maîtrisés. La quatrième couche montre si un élève a marqué un concept comme déjà vu.

La mise à jour des données de chaque couche est faite de façon indépendante. Par exemple, si dans un formulaire un élève ne marque pas une option parce qu'il pense qu'il n'a pas suffisamment de connaissances sur le concept, le matériel portant sur ce concept restera accessible et l'étudiant peut y revenir.

### **2.3.3 Leactivemath**

LeActivemath (2008, [www.levivemath.org](http://www.levivemath.org)) est un système web intelligent de « troisième génération » pour apprendre les mathématiques

**Le Math Active** Main Page

Hello Anonymous.  
This is the main page of ActiveMath. Please choose one of the books.  
To use and see all features of ActiveMath, you have to [login](#).

**Prerecorded Books**

**English Books:**

- **Complete Content of LeAM\_calculus**   
Number of pages: 220
- **Highschool Content of LeAM\_calculus**   
Number of pages: 166
- **LeActiveMath Grade 11**   
Number of pages: 40
- **Secondary School Content of LeAM\_calculus**   
Number of pages: 160
- **University Content of LeAM\_calculus**   
Number of pages: 192
- **Zoology of exercises in LeAM\_calculus**   
Number of pages: 34

**Personal Books**

**No books available.**

With ActiveMath, you can create personal books to suit your needs. Just try it!

To use this feature, you have to [login](#) first.

Welcome to the evaluation server of the [LeActiveMath](#) research project.

To use LeActiveMath, you need:

- [Firefox](#), at least version 1.5, or Internet Explorer 6 (other browsers may work, but are not our primary target). Cookies and Javascript need to be enabled.
- [Java Plugin](#) for your browser
- screen size: 1024x768 or better.
- a fast Internet connection (DSL).

Server updates will be installed daily between 12:30 and 1:30 pm GMT. You might be experiencing server downtimes during this period. We are sorry for the inconvenience.

Figure 8 Page principale de LeActiveMath ([www.leactivemath.org/system-architecture.html](http://www.leactivemath.org/system-architecture.html))

Il s'agit d'un projet de recherche qui combine la technologie web avec les didacticiels intelligents et la technologie du langage naturel. LeActiveMath présente l'information du cours de mathématique de façon adaptative selon :

- Les buts des étudiants
- Le scénario d'apprentissage
- Le niveau individuel de connaissance de chaque étudiant
- Les préférences des utilisateurs

LeActiveMath utilise un moteur de recherche sémantique, un outil de carte graphique de concepts, un outil d'évaluation et une banque de données d'exercices.

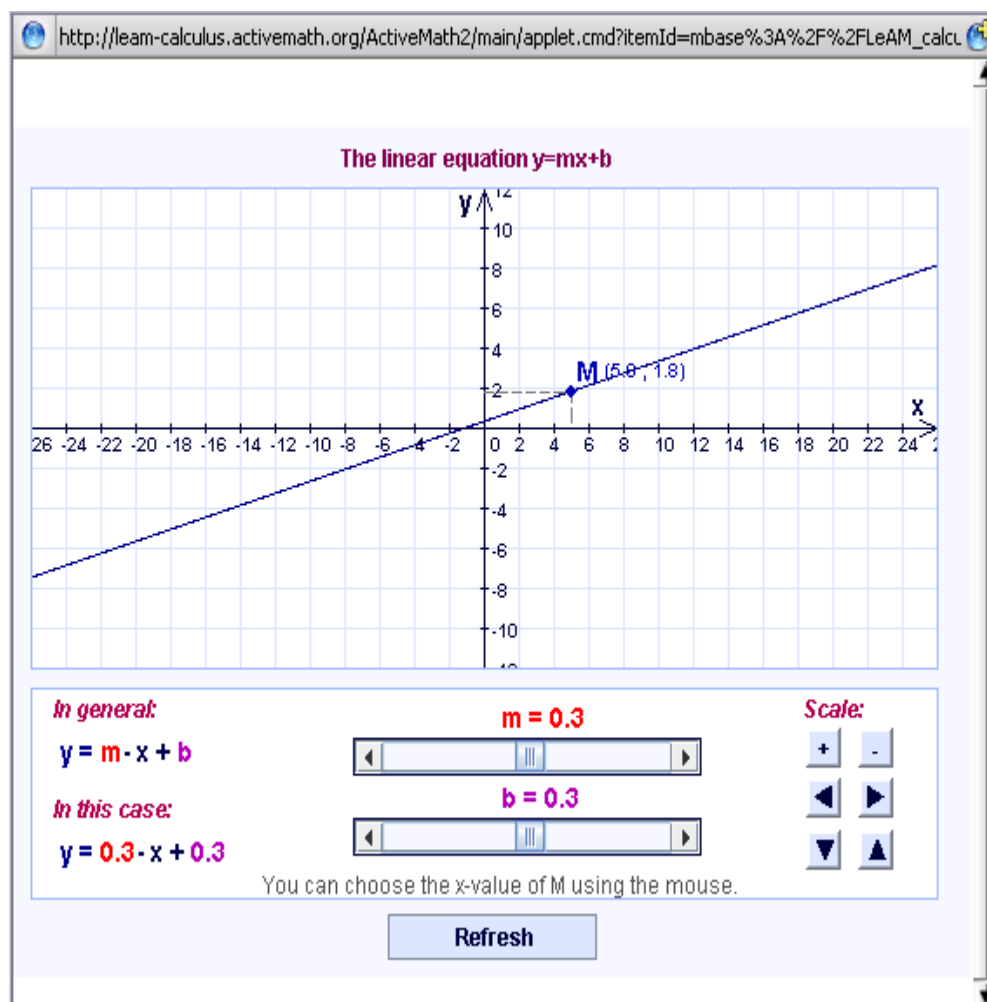


Figure 9 Outil interactif pour travailler avec les équations linéaires  
 (www.leactivemath.org/system-architecture.html)

Les élèves peuvent configurer la quantité d'exercices interactifs, le type de dialogues dans le tuteur et les caractéristiques d'apprentissage. Ils peuvent choisir les objets d'apprentissage à ajouter dans les exercices et analyser son modèle d'apprentissage et faire des modifications. En plus, le système peut adapter l'interface de l'utilisateur selon l'historique de l'élève.

Le modèle d'apprentissage vise à identifier les facteurs suivants :

- les valeurs de facteurs tels que la motivation et l'attitude.



- les sujets et exercices maîtrisés.

Du côté pédagogique le système utilise :

- une approche constructiviste.
- un soutien à l'apprentissage basé sur des problèmes.
- un modèle du niveau de connaissances.
- un modèle de connaissances pédagogiques de huit modules (Figure 10) :



le système travaille avec un modèle des utilisateurs dans l'environnement d'apprentissage intelligent.

Nous avons vu aussi quelques exemples commerciaux des applications de guidage et des applications des tutoriels. Toutes les applications essaient de connaître le niveau de connaissances des utilisateurs pour orienter le type d'aide à offrir aux utilisateurs. Nous avons vu que toutes les applications utilisent les tests adaptatifs pour connaître le niveau de connaissance de l'étudiant. Les tests adaptatifs sont effectivement un outil très populaire dans ces applications pour créer un modèle de connaissance de l'utilisateur. Dans le prochain chapitre, nous allons voir une description plus détaillée des tests adaptatifs.

### CHAPITRE 3 MODÈLES DE LA CONNAISSANCE

Le chapitre précédent décrit différentes applications qui reposent sur un modèle de la connaissance pour fournir une forme de soutien pédagogique. Ce chapitre décrit l'approche traditionnelle pour modéliser et diagnostiquer la connaissance d'un individu. Cette approche est la TRI (Théorie des Réponses aux Items). Elle a été élaborée dans le cadre des tests adaptatifs (Computer Adaptive Testing) et raffinée depuis environ trois décennies. Cette description est suivie de celle d'une approche alternative à la modélisation de la connaissance basée sur la théorie des espaces de connaissances, et dont le développement est plus récent (Falmagne et Doignon, 1998). La théorie des espaces de connaissances sert de base au modèle utilisé dans cette étude et qui fait l'objet du prochain chapitre.

#### 3.1 Modèle de la Théorie des Réponses aux Items

L'approche de la Théorie des Réponses aux Items (TRI) est issue des applications de tests adaptatifs (abréviation en anglais de Computing Adaptive Testing) qui sont fréquemment utilisées dans le cadre de tests bien connus comme le GMAT, le TOEFL, le GRE, etc., ainsi que plus récemment dans le cadre des environnements intelligents d'apprentissage.

La Théorie des réponses aux items est une approche basée sur l'apprentissage : le modèle pour un domaine de connaissance est calibré à partir de données dichotomiques de tests (Millan et Perez-de-la-cruz, 2002; Mislevy et Almond, 1997). Dans la TRI, la compétence de l'individu est représentée par  $\theta$  et on présume que le succès aux items  $X$  est déterminé par cette compétence. De plus, on présume que les réponses sont indépendantes entre elles, mais dépendent des paramètres qui représentent les caractéristiques des items, comme par exemple la difficulté et la discrimination (sensibilité du succès aux compétences d'un utilisateur).

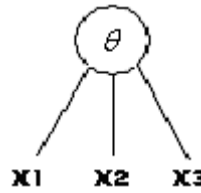


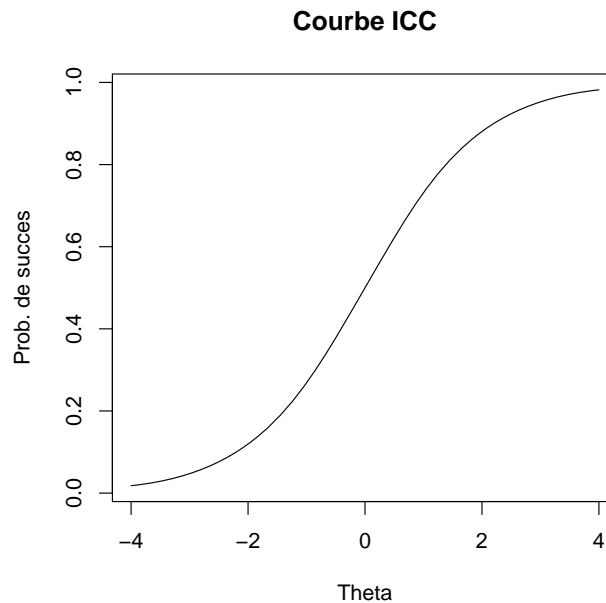
Figure 11 La structure du modèle TRI

La TRI présume qu'étant donné  $\theta$ , le succès à un item  $X_i$  est indépendant des autres items  $X_2$  et  $X_3$ . La probabilité qu'un utilisateur réponde correctement à une question  $X_i$  est entièrement déterminée par  $\theta$ .

Or, le lien entre  $\theta$  et la probabilité de succès à un item  $X_i$  est défini par la fonction ICC (courbe caractéristique de l'item) (Garrido, 2006). Chaque item (question) du test possède une ICC spécifique. Le graphique ICC est une courbe en S, tel qu'illustré à la Figure 12. On peut constater que plus l'habileté  $\theta$  augmente, plus la probabilité de succès augmente. La formule de cette courbe est la suivante :

$$P(X_i | \theta) = \frac{1}{1 + e^{-a(\theta + b)}} \quad (1)$$

où  $a$  est la *discrimination* et influence la pente de la courbe ICC, tandis que  $b$  est la difficulté et influence le point d'inflexion de la courbe (fixé à  $b=0$  dans la Figure 12)



**Figure 12 Exemple du graphique d'une courbe ICC**

Si l'évaluation des chances de succès à un item  $X_i$  étant donné  $\theta$  est dérivée directement de la formule (1), l'évaluation de  $\theta$  à partir d'un ensemble de réponses est obtenue par approximation numérique.

Prenons l'exemple suivant pour illustrer ce calcul : le modèle de Rasch pour  $n$  items observés. Ce modèle est basé sur la formule suivante :

$$P(X_1, \dots, X_n | \theta, \beta_1, \dots, \beta_n) = \prod_{j=1}^n P(X_j | \theta, \beta_j) \quad (2)$$

où

$X_j$  est la réponse pour l'item  $j$ , soit 1 pour une réponse correcte et 0 pour une erreur.

$\beta_j$  est le paramètre de difficulté de l'item  $j$  (le  $b$  de la formule (1))

$\beta_1, \dots, \beta_n$  sont les valeurs pour  $\beta$  obtenues à partir d'une grande quantité d'échantillons de réponses des utilisateurs.

$X_1, \dots, X_n$  sont les vecteurs de réponses et les valeurs de probabilité de  $\theta$ .

Comme dans tous les modèles TRI, cette formule repose sur l'indépendance des observations  $P(X_i | \theta, \beta_i)$ . Le principe consiste à trouver la valeur maximale de  $P(X_1, \dots, X_n | \theta, \beta_1, \dots, \beta_n)$  à partir du produit des  $P(X_i | \theta, \beta_i)$  individuels. Or, la valeur de  $\beta_i$  correspond à  $b$  dans l'équation (2) et elle est déterminée empiriquement à partir des données d'entraînement. En général, plusieurs dizaines de données et jusqu'à quelques centaines de données sont nécessaires pour obtenir une valeur fiable. On peut donc varier la valeur de  $\theta$  afin de maximiser le terme de gauche de l'équation (2) et ainsi obtenir la valeur de  $\theta$  la plus élevée, c'est-à-dire la plus probable étant donné les observations. Ce calcul est fait par des méthodes numériques (Baker et Kim, 2004). La valeur de  $\theta$  est évaluée à la suite de chaque réponse pour établir le choix du prochain item. Différentes techniques existent pour calculer l'item qui apportera le plus d'information afin d'estimer  $\theta$  avec le plus de précision possible tout en minimisant le nombre d'items, comme par exemple le critère de l'information maximale Fisher (Baker, 2001).

### 3.2 Les espaces de connaissances

Une autre approche à la modélisation des connaissances est la théorie des espaces de connaissances de Falmagne et Doignon (1998). Cette approche à la modélisation est utilisée dans le système ALEKS (voir chapitre 2) et elle sert aussi de base à l'approche POKS qui est utilisée pour la présente étude et que nous verrons au chapitre qui suit.

Cette théorie stipule que la maîtrise des items de connaissance (les items d'un test ou des exercices) s'effectue dans un ordre partiellement ordonné. La Figure 13 démontre le principe pour quatre items de connaissance,  $\{a, b, c, d\}$ . L'état de connaissance d'un individu (le diagnostic de ses compétences) est représenté par un sous ensemble de

l'ensemble des quatre items. On y retrouve 7 états de connaissances possibles et une séquence d'ordonnement à travers ces items, ce que l'on nomme un espace de connaissances. Ainsi, un individu apprend en premier 'a', puis 'b' ou 'c' mais sans contrainte d'ordre entre ces deux, puis 'd', puis finalement le dernier item non appris qui sera soit 'c', soit 'b' selon le cas. Ces contraintes d'ordre d'apprentissage peuvent se représenter par un graphique ET-OU et se caractérisent par la fermeture des états de connaissances sous l'union et l'intersection (Falmagne et Doignon, 1999). La Figure 14 illustre la structure de connaissance représentée par un graphe ET-OU équivalent à l'espace de connaissance de la Figure 13. Dans ce formalisme, les liens de 'd' vers 'b' et 'd' vers 'c' sont une disjonction et signifient qu'un ou l'autre peut être satisfait. C'est-à-dire que l'on peut apprendre 'b' ou 'c' avant d'apprendre 'd', mais un seul des deux nœuds prérequis est nécessaire. Formellement, un graphe ET-OU permet de représenter tout espace de connaissance fermé sous l'intersection et l'union.

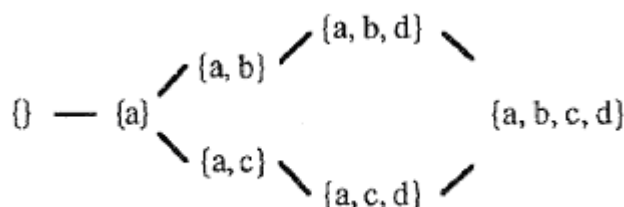


Figure 13 Espace de connaissances. Chaque sous-ensemble représente un état possible de la connaissance d'un individu.



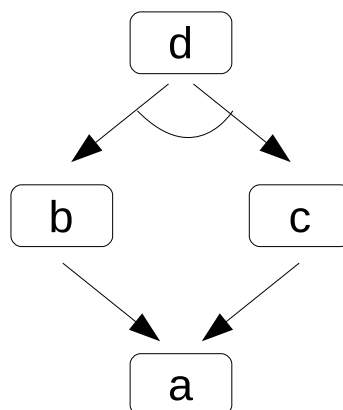


Figure 14. Structure de connaissances représentant l'ordre d'apprentissage équivalent à la Figure 13.

Comme mentionné, la théorie des espaces de connaissances est utilisée dans le système ALEKS (section 2.2.1). Elle a l'avantage de préciser l'ordre d'apprentissage. De cet ordre, il est possible de déterminer ce que l'apprenant devrait apprendre dans la prochaine étape. Par exemple, on peut voir de la Figure 14 qu'un apprenant qui maîtrise l'item 'b' serait prêt à apprendre l'item 'd' ou l'item 'c' (à cause de la relation ET-OU). De même, un apprenant qui maîtrise l'item 'a' serait prêt pour les items 'b' ou 'c', mais pas pour l'item 'd' car soit 'c', soit 'b' doivent être appris au préalable.

Une version plus contrainte de cet ordre d'apprentissage est utilisée dans l'approche POKS où les contraintes d'apprentissage peuvent se représenter par un graphe orienté du type ordre partiel (Figure 15). POKS sera décrit en détail dans le prochain chapitre mais nous faisons le lien entre ce modèle et la théorie des espaces de connaissances dans un premier temps.

Pour le formalisme de POKS, l'espace de connaissance est fermé uniquement sur l'union et non pas sur l'union et l'intersection comme la théorie des espaces de connaissances le stipule. Cette contrainte est donc plus forte que pour les espaces de connaissances, mais elle permet de découvrir la structure des connaissances avec un nombre réduit de données et s'avère donc beaucoup plus pratique en réalité.

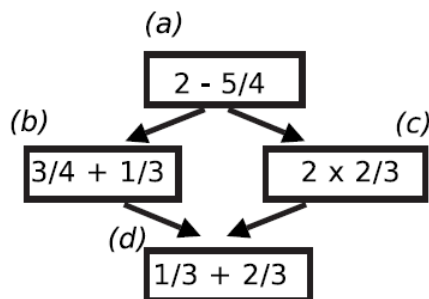


Figure 15 Exemple des liaisons des items dans un espace de connaissances

Dans la Figure 15, nous pouvons observer les liens suivants qui indiquent l'ordre d'apprentissage :

"a"  $\rightarrow$  "b"  $\rightarrow$  "d"

"a"  $\rightarrow$  "c"  $\rightarrow$  "d"

Il n'y a pas d'ordre d'apprentissage entre "b" et "c". Les états de connaissances possibles correspondent à un sous-ensemble dans l'espace de connaissance :  $\{\{\}, \{d\}, \{b,c\}, \{c,b\}, \{b,c,d\}, \{a,b,c,d\}\}$ . Ces liens permettent ainsi d'effectuer des inférences et un diagnostic de connaissance. Si un étudiant a réussi à répondre la question "a", la probabilité de réussir les questions "b", "c" et "d" sera plus élevée. Si un étudiant a échoué à la question "d", la probabilité de réussir les questions "a", "b" et "c" sera plus faible.

Dans la théorie des espaces de connaissances, comme pour POKS, la modélisation de la connaissance d'un individu se représente par un sous-ensemble des items. Il n'existe donc pas de concepts ou autres compétences dites cachées. Toute la connaissance modélisée correspond à une performance observable comme la réussite à des items d'un test ou à des exercices, ou encore à l'exécution d'une opération avec succès ou de toute autre manifestation observable de compétences. Pour déterminer si une compétence de plus haut niveau est acquise, on peut toutefois utiliser les items observables, comme on le

fait dans un examen où l'on regroupera un ensemble d'items pour décider si un sujet est maîtrisé ou non.

Le fait de ne modéliser que des compétences observables facilite l'entraînement du modèle puisqu'il n'y a pas de variables cachées, non observables. En effet, plus il existe de variables cachées dans un modèle, plus l'entraînement nécessite une grande quantité de données. Cette approche contraste avec celle des réseaux bayésiens qui intègrent souvent de nombreux concepts, qui sont des nœuds cachés car on ne peut observer directement si un concept est maîtrisé ou non (voir par exemple Vomlel, 2004).

## CHAPITRE 4 LE MODÈLE POKS

La théorie des espaces de connaissance décrite dans la section qui précède est la base du modèle POKS (Partial Order Knowledge Structure) (Desmarais et Pu., 2005; Desmarais, Maluf et Liu, 1995) qui est utilisé dans la présente étude. La Figure 15 représente en fait une structure POKS. Cette structure est un graphe acyclique qui reflète l'ordre d'apprentissage des items. Nous décrivons dans ce chapitre comment une telle structure peut être construite à partir de données de tests et comment elle est utilisée pour effectuer des inférences et un diagnostic des compétences d'un individu.

### 4.1 Définitions

Conformément à la théorie des espaces de connaissances dont le modèle POKS dérive, l'état de connaissance d'un individu correspond à un diagnostic de ses compétences et il est représenté par un sous-ensemble d'items de connaissances.

Définissons l'ensemble des items d'un domaine de connaissance par  $\Psi$ , où chaque item est représenté par  $X_i$  :

$$\Psi \equiv \{X_1, X_2, \dots, X_n\}$$

Chaque item peut représenter une compétence observable comme un item dans un questionnaire ou une compétence manifestée dans le comportement d'un individu. Chaque item peut prendre la valeur « maîtrisé » ou « non maîtrisé ». Un état de connaissance d'un individu  $j$ , sera représenté par un sous-ensemble,  $S_j$ , du domaine de connaissance :

$$S_j \subseteq \Psi$$

Contrairement à la théorie des espaces de connaissances, pour le modèle POKS, chaque item est assigné une probabilité de maîtrise. On désigne la probabilité que l'individu  $j$  maîtrise l'item  $i$  par :

$$P(X_{ji}) : \text{probabilité que } j \text{ maîtrise l'item } i$$

À noter que, à moins que le contexte l'exige, nous éviterons d'utiliser l'indice  $j$  dans  $P(X_{ji})$  et utiliserons  $P(X_i)$  pour référer à la probabilité qu'un individu maîtrise l'item  $i$ .

Ainsi, pour le modèle POKS, la définition d'un état de connaissance (ou un diagnostic de compétences) est donc :

$$S_j \equiv \{P(X_1), P(X_2), \dots, P(X_n)\}$$

## 4.2 Les liens

Comme mentionné précédemment, le modèle POKS repose sur une structure des connaissances qui forme un graphe dirigé acyclique qui reflète l'ordre d'apprentissage des items et il a la propriété de définir des sous-ensembles plausibles d'états de connaissances qui sont fermés sous l'union, mais pas sous l'intersection comme le stipule la théorie des espaces de connaissances. Cette contrainte supplémentaire réduit la quantité de données nécessaires pour induire la structure (Desmarais et Pu, 2005).

La construction de la structure se fait à partir de réponses d'un échantillon de sujets à un test comportant des items de connaissance qui représentent les noeuds de la structure. Le processus consiste en des tests d'hypothèses statistiques effectués pour chaque paire d'items afin de déterminer si une relation existe entre les deux.

Par exemple, prenons la distribution des bonnes et mauvaises réponses d'un ensemble de répondants à un tests pour deux items,  $X_a$  et  $X_b$ . Si on a une relation  $X_a \rightarrow X_b$ , alors on peut s'attendre à trouver des répondants dans chacune des trois premières conditions ( $\{x_{ab}, x_{-ab}, x_{-a-b}\}$ ) mais pas pour la quatrième ( $x_{a-b}$ ) :

Tableau 1. Tableau de contingence des possibilités de combinaisons des items  $X_a$  et  $X_b$

Condition		$X_a$	$X_b$	Répondants
(1)	$x_{ab}$	réussi	réussi	oui
(2)	$x_{\neg a \neg b}$	échoué	échoué	oui
(3)	$x_{\neg ab}$	échoué	réussi	oui
(4)	$x_{a \neg b}$	réussi	échoué	non

En effet, étant donné la relation  $X_a \rightarrow X_b$ , on ne s'attend pas à trouver aucun répondant qui réussit  $X_a$  mais qui échoue l'item normalement plus facile,  $X_b$ .

Les liens entre les unités de connaissance peuvent être considérés comme des relations d'implication  $X_a \rightarrow X_b$ , c'est-à-dire : si  $X_a$  est connu alors  $X_b$  doit être connu, c'est-à-dire  $P(X_b|X_a)=1$ , ou si  $X_b$  n'est pas connu donc  $X_a$  n'est pas connu, c'est-à-dire  $P(X_{\neg a}|X_{\neg b})=1$ . Pour vérifier que le lien se trouve supérieur à un seuil minimal donné, il faut tester les deux hypothèses ci-haut, en utilisant les inégalités :

$$P\left([P(X_b|X_a) \leq p_c] \mid D\right) \leq \alpha_c \quad (3)$$

$$P\left([P(X_{\neg a}|X_{\neg b}) \leq p_c] \mid D\right) \leq \alpha_c \quad (4)$$

$p_c$  : probabilité conditionnelle minimale.

$\alpha_c$  : erreur alpha de la probabilité conditionnelle minimale.

$D$  : fréquence de distribution de co-occurrences de  $X_a$  et  $X_b$  dans la banque de données.

L'inégalité (3) stipule que la probabilité conditionnelle du succès de  $X_b$  étant donné le succès de  $X_a$  doit être plus petite ou égale à un seuil  $p_c$ . Nous pouvons obtenir cette conclusion à partir des données de l'échantillon de données  $D$  et d'un calcul à partir d'une distribution binomiale cumulative. Dans l'inégalité (3), la valeur de

$P([P(X_b | X_a)] | D)$  est obtenue par la sommation de la fonction binomiale pour toutes les distributions où  $x_{a \rightarrow b}$  est plus petit que la fréquence observée en  $D$ , c'est-à-dire :

$$\begin{aligned} P([P(X_b | X_a)] | D) &= P(x \leq x_{a \rightarrow b} | X_a) \\ &= \sum_{i=0}^{x_{a \rightarrow b}} Bp(i, x_a, p_c) \\ &= \sum_{i=0}^{x_{a \rightarrow b}} \binom{x_a}{i} p_c^i (1 - p_c)^{x_a - i} \end{aligned}$$

où  $x_a = x_{ab} + x_{a \rightarrow b}$ . La probabilité conditionnelle de la seconde condition (inégalité 4) repose sur la même fonction, mais utilise  $Bp(i, x_{-b}, p_c)$  plutôt que  $Bp(i, x_a, p_c)$ .

De plus, nous avons besoin d'appliquer un troisième test pour vérifier que les probabilités conditionnelles sont différentes des probabilités initiales. Il faut appliquer :

$$P(X^2) < \alpha_i \quad (5)$$

$\alpha_i$  : erreur alpha de l'interaction.

Un réseau POKS se construit en comparant toutes les paires d'items possibles pour déterminer si elles passent les trois tests. Si les trois tests entre les items  $X_a$  et  $X_b$  sont positifs pour la distribution  $D$  (inégalités (3), (4) et (5)), on conclura alors qu'il y a une relation  $X_a \rightarrow X_b$ . Le choix des seuils de  $p_c$  et de  $\alpha_c$  détermine la force des relations retenues. Par la suite, chaque relation aura un poids qui permettra de déterminer la force de l'influence d'un item sur l'autre et de mettre à jour les probabilités à la lumière des observations.

### 4.3 Actualisation des probabilités

Une fois une structure POKS définie, elle détermine les interrelations entre les items et permet l'inférence de l'état de connaissance d'un individu à partir de l'observation d'items réussis ou échoués. Ce diagnostic est basé sur l'actualisation des probabilités que nous décrivons dans cette section.

### 4.3.1 Sélection des nœuds évidence

Le modèle POKS vise à prédire le succès pour chaque item non observé basé sur les items observés. Chaque item dans la structure POKS peut donc représenter soit un nœud « *évidence* »,  $X_e$ , soit un nœud dont on tente d'estimer la probabilité de maîtrise,  $X_t$ , c'est-à-dire qu'on tente d'estimer la probabilité conditionnelle :  $P(X_t | X_e)$ . Évidemment, un nœud estimé devient un nœud *évidence* dès qu'il est observé.

Ce ne sont toutefois pas tous les nœuds qui peuvent servir « *d'évidence* » pour prédire la probabilité de maîtrise d'un nœud,  $X_t$ . Pour chaque item dont on tente d'estimer la probabilité, il faut donc déterminer quels sont les nœuds pertinents. C'est ce qui est fait lorsqu'on construit la structure POKS. Ainsi, tous nœuds avec lesquels  $X_t$  passe les tests des inégalités (3), (4) et (5) sont considérés des nœuds *évidence*. On obtient donc, pour chaque item, une équation avec un ensemble d'items servant *d'évidence*,  $\mathbf{X}_e$  :

$$P(X_t | \mathbf{X}_e) = P(X_t | X_1, X_2, \dots, X_k) \quad \{\forall X_t \in \Psi\}$$

où  $\{X_1, X_2, \dots, X_k\}$  est l'ensemble des items considérés pertinent pour estimer la probabilité de  $X_t$ .

### 4.3.2 Probabilité postérieure

Le modèle POKS repose sur l'hypothèse d'indépendance entre les nœuds *évidence*. Cette hypothèse s'exprime par la simplification suivante :

$$P(X_i, X_j | X_t) = P(X_i | X_t) \cdot P(X_j | X_t) \quad \{X_i, X_j, X_t \in \Psi\}$$

Étant donné cette hypothèse, la règle de Bayes peut être réécrite sous la forme suivante :

$$P(X_t | \mathbf{X}_e) = \frac{P(X_t)}{P(X_1, X_2, \dots, X_k)} \prod_i^k P(X_i | X_t)$$



où  $\mathbf{X}_e$  est l'ensemble des  $k$  nœuds évidence  $\{X_1, X_2, \dots, X_k\}$

Ce calcul correspond en fait à celui d'une probabilité postérieure sous l'hypothèse d'indépendance des *évidences* (observations). Dans POKS, il est réalisé en utilisant les ratios de chance. Pour mettre à jour la probabilité de chaque item de connaissance en regard des items réussis ou échoués, on utilise un algorithme de calcul de probabilité postérieures en utilisant la vraisemblance de suffisance (LS) et la vraisemblance de nécessité (LN).

$$LS_{a \rightarrow b} = \frac{O(X_b | X_a)}{O(X_b)} \quad (6)$$

$$LN_{a \rightarrow b} = \frac{O(X_a | \neg X_b)}{O(X_a)} \quad (7)$$

où la définition de  $O(X)$  est celle de la chance (*odds* en anglais) :

$$O(X) = P(X) / Q(X)$$

$$Q(X) = P(\neg X) = 1 - P(X)$$

$$O(X | Y) = P(Y | X) / Q(X | Y)$$

Chaque fois qu'un nœud change sa valeur de probabilité, il faut faire la propagation de l'évidence pour tous les nœuds connectés du réseau. Si on observe que  $X_a$  est vrai, alors la probabilité de  $X_b$  sera actualisée en utilisant l'équation :

$$O(X_b | X_a) = LS_{a \rightarrow b} O(X_b)$$

et si on observe que  $X_a$  est faux, l'actualisation se fait selon l'équation :

$$O(X_a | \neg X_b) = LN_{a \rightarrow b} O(X_a)$$

Pour les petits échantillons, il est nécessaire, lorsqu'on estime ces probabilités à partir de données de fréquences, d'effectuer une correction dite de Laplace. Dans le cas où l'on a que deux catégories possibles (vrai et faux) et où l'estimé a priori de la probabilité de chaque classe est de 0,5, alors cette correction consiste à ajouter la valeur

de 1 au numérateur et de 2 au dénominateur (Margineantu et Diettrich, 2002). Par exemple, une observation de la fréquence de  $X=succès$  de 0/1 (une seule observation correspondant à un échec) est corrigée avec le ratio  $(0+1)/(1+2)=1/3$ . L'absence d'observations se traduit aussi en une valeur de 1/2. La correction de Laplace est appliquée dans notre étude pour toute probabilité estimée à partir de la fréquence d'occurrence dans les échantillons.

#### **4.4 Résumé**

Dans ce chapitre, nous avons vu les caractéristiques principales de POKS et les bénéfices de son utilisation pour construire le modèle de connaissances, à savoir la simplicité des calculs qui se reflète sur la performance en temps de calcul pour les inférences et sur la faible quantité de données nécessaires à l'entraînement. POKS est l'approche adoptée pour faire l'inférence de l'état de connaissance d'un étudiant à partir des données.

## CHAPITRE 5 MÉTHODOLOGIE

### 5.1 Retour sur la problématique et les objectifs

L'approche POKS nous donne la possibilité de faire un diagnostic de connaissance à partir d'un petit nombre d'items. Le réseau graphique de POKS est dérivé de données d'entraînement obtenues à partir des tests que les élèves ont faits. Malgré l'efficacité de l'approche quant à la quantité de données nécessaires, le recueil des données pour des tests de plusieurs dizaines, voire des centaines de questions, pose un problème. Ce n'est pas pratique de demander à quelques utilisateurs de répondre à un test contenant plus de cent questions. Il faut donc recourir à l'entraînement du réseau POKS avec données partielles.

Les données partielles peuvent toutefois entraîner une dégradation de la performance du modèle induit. Cependant, dans la mesure où l'on contrôle la façon dont on échantillonne les données, on peut se questionner sur l'impact du choix des items à échantillonner et s'il est possible de choisir les items de façon à augmenter la performance du modèle créé à partir des échantillons de données partielles.

On pourrait, par exemple, faire l'hypothèse que les items qui sont reliés à plusieurs autres items dans un réseau POKS ont une influence plus importante que ceux qui ont moins de liens, et qu'ainsi il soit préférable de leur accorder un plus grand nombre d'observations afin que leur calibration soit plus précise. Bien qu'on ne puisse pas connaître la structure du réseau au préalable, on sait que les items les plus difficiles auront un faible taux de succès et les items les plus faciles auront un taux de succès élevé. Ces items se retrouveront respectivement près des racines et près des « feuilles » (nœuds terminaux) du réseau. Les items de difficulté moyenne se retrouveront plutôt au milieu d'un réseau POKS. En termes d'entropie, nous dirions que les items de haute entropie seront au milieu du réseau (difficulté moyenne) alors que les items de faible entropie se retrouveront aux racines et aux feuilles du réseau.

Typiquement dans une structure POKS, les nœuds au milieu ont plus de relations avec les autres nœuds que ceux à la racine ou que les nœuds terminaux. D'où l'hypothèse que ce sont les nœuds à haute entropie qui sont les plus susceptibles d'amener le plus d'information pour le diagnostic en général et qu'un échantillonnage biaisé en leur faveur peut fournir un modèle plus performant qu'un échantillonnage aléatoire. Ce sont les questions qui seront explorées.

L'objectif de ce projet est de déterminer si on peut définir une méthode d'échantillonnage de données partielles qui permet d'obtenir un modèle POKS plus performant qu'une méthode d'échantillonnage aléatoire.

L'objectif plus spécifique de cette étude est d'examiner l'évolution de la performance prédictive du modèle graphique POKS en fonction des paramètres d'échantillonnage.

Les paramètres d'échantillonnage étudiés sont basés sur l'entropie des items, à savoir les items à haute entropie (dont le taux de succès est aux environs de 0,5) et à basse entropie (taux de succès près de 0 ou de 1). Nous visons à déterminer si un échantillonnage qui favorise les items à haute ou à basse entropie peut influencer la performance prédictive du modèle bayésien POKS. Si les résultats sont positifs, des études ultérieures permettront de déterminer si les conclusions se généralisent à d'autres modèles.

## **5.2 Conditions expérimentales**

Cette étude explore l'effet de différents schèmes d'échantillonnage des données sur la performance du modèle de diagnostic des connaissances.

Un premier schème consiste en un échantillonnage aléatoire d'un sous-ensemble des données complètes. Ce schème servira de point de comparaison.

Un second schème consiste à favoriser l'échantillonnage des items comportant la plus forte entropie ( $p=0,5$ ). En pratique, il est plausible d'établir d'abord le taux moyen de

succès des items avec un petit nombre de données, pour ensuite guider l'échantillonnage afin de favoriser un plus grand nombre d'items de difficulté moyenne.

Le troisième schème consiste à favoriser plutôt les items de faible entropie, c'est-à-dire de difficulté grande ou faible. Ici encore, ceci est possible en pratique en utilisant préalablement un petit échantillon pilote qui servira à guider l'échantillonnage par la suite.

Selon ces schèmes, trois conditions expérimentales sont définies :

- Échantillons uniformes : même probabilité de sélection pour tous les items.
- Échantillons plus incertain : probabilité de sélection plus élevée pour les items incertains.
- Échantillons moins incertain : probabilité de sélection plus basse pour les items incertains.

Rappelons que les nœuds incertains, ou de difficulté moyenne, sont ceux qui ont une grande entropie et un taux de succès près de 0,5, alors que les nœuds les moins incertains sont de faible entropie et de difficulté faible ou forte, et que leur taux de succès avoisine 0 ou 1.

### **5.3 Méthodes d'échantillonnage**

La variation d'échantillonnage est basée sur une distribution de probabilité d'échantillonnage des items. Le principe consiste à assigner la probabilité qu'un item soit échantillonné en fonction de son niveau d'incertitude. La Figure 16 illustre cette distribution. Elle représente le segment  $[0, 2,5]$  de la distribution Gaussienne assignée en fonction du rang de l'item, où le rang est calculé en fonction de l'entropie. Dans le cas des échantillons les plus incertains, le rang est basé sur l'entropie avec la valeur la plus élevée ayant comme rang 1, de sorte que les items de difficulté moyenne auront une forte probabilité d'être échantillonné (près de 0,4), tandis que les items de faible entropie

auront une probabilité aux environs de 0,02, comme on peut le voir sur la Figure 16. Dans les cas des échantillons les moins incertains, les rangs sont inversés et ce sont les items difficiles et faciles qui auront la plus grande probabilité d'être échantillonnés.

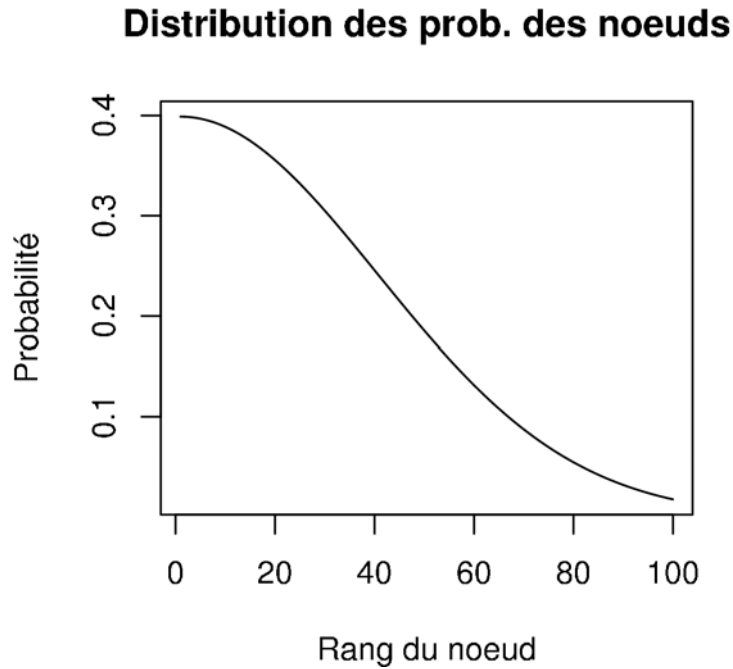


Figure 16 Distribution de la probabilité d'échantillonnage des items en fonction de leur difficulté (rang en percentile)

L'algorithme pour créer les différents échantillons est détaillé dans la section qui suit.

### 5.3.1 Algorithme d'échantillonnage

Définition  $Sp$  une fonction qui retourne échantillon de données de  $\Psi$  où  $n$  représente la taille de l'échantillon :

$$Sp(\Psi, n) \subset \Psi \quad (n < |\Psi|)$$

Et définissons  $f(X)$  comme une distribution de probabilité de  $X$  sur  $\Psi$ , c'est-à-dire la distribution des probabilités d'échantillonnage de chaque item  $X$  dans  $\Psi$ . À noter que nous écrivons  $X$  en place de  $X_i$ , en omettant l'indice  $i$  pour alléger la notation.

Posons  $f_u(X)$  comme la distribution rectangulaire définie par :

$$f_u(X) = 1/m$$

où  $m$  représente le nombre d'items dans  $\Psi$ , i.e.  $m = |\Psi|$ .

Pour les échantillons non uniformes, la distribution de probabilité sera une fonction de l'incertitude des items. Chaque item est assigné un rang basé sur son niveau d'incertitude et mesuré par l'entropie :

$$E(X) = P(X)\log(P(X)) + (1-P(X))\log(1-P(X))$$

Plus  $E(X)$  est élevé, plus  $X$  est incertain. La valeur de  $P(X)$  est estimée à partir d'un échantillon et correspond au taux de succès de  $X$ . Cette estimation est calculée en appliquant la correction de Laplace :

$$P(X) = \frac{n+1}{n+m+2}$$

où  $n$  est le nombre d'individus qui ont réussi l'item  $X$  et  $m$  est le nombre d'individus qui ont échoué.

Étant donné le rang donné par l'entropie, la probabilité assignée à chaque item est fonction de ce rang.

Définissons deux distributions de probabilité,  $f_{v1}(X)$  et  $f_{v2}(X)$ , représentant deux schèmes d'échantillonnage non uniforme basé sur le une fonction rang,  $rank(X)$ , qui retournent le rang relatif de  $X$  dans l'ensemble  $\Psi$  et ordonné par l'entropie de l'item. En définissant  $rank_1(X)$  comme l'ordre décroissant d'entropie et  $rank_2(X)$  comme l'ordre croissant, alors, on peut assigner la probabilité qu'un item  $X$  soit échantillonné

basé sur le rang en fonction d'une section de la distribution normale (gaussienne) et ainsi définir  $f_{v1}(X)$  comme :

$$f_{v1}(X) = K \cdot \varphi(b \cdot \text{rank}_1(X)/m)$$

et  $f_{v2}(X)$  comme :

$$f_{v2}(X) = K \cdot \varphi(b \cdot \text{rank}_2(X)/m)$$

où la fonction  $\varphi(x)$  est la distribution normale avec  $\mu = 0$  et  $\sigma = 1$ , et qui correspond à :

$$\varphi(x) = \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi}}$$

et où:

- $m = |\Psi|$
- $b$  est un point arbitraire qui spécifie la probabilité minimale d'échantillonnage d'un item.
- $K$  est une constante de normalisation qui assure que la surface sous la courbe de distribution des probabilités soit de 1.  $K$  correspond donc à l'intégration de la fonction  $\varphi(x)$  entre 0 et  $b$  :

$$K = \int_0^b \varphi(x) d(x)$$

Pour notre expérience, la valeur choisie pour  $b$  est fixée à 2,5 et correspond à une probabilité minimale de 0,04. Le graphique de la Figure 16 correspond à la densité de probabilité de  $f_{v1}(X)$  pour  $b = 2,5$ . Pour la densité de probabilité fournie par  $f2(X)$ , l'ordre est alors inversé et la courbe est alors symétrique, mais inversée par rapport à la Figure 16.

Il faut noter que le choix de la fonction  $\varphi(x)$  pour établir la distribution de probabilité de  $X$  en fonction de l'entropie (du taux de succès) est arbitraire. Une fonction sigmoïde aurait tout aussi bien pu être utilisée et d'autres fonctions ainsi que d'autres paramètres pourraient être explorés dans le futur.



#### 5.4 Simulations et mesure de la performance d'un modèle POKS

Des structures POKS sont construites à partir des échantillons décrits dans le paragraphe précédent et selon la technique décrite dans le chapitre 4. Une structure POKS représente un modèle pour lequel nous voulons mesurer la performance prédictive. Cette performance est mesurée à travers un processus de simulation de test adaptatif, où une question est choisie et la réponse utilisée pour recalculer les probabilités de succès de chaque autre item qui n'a pas encore été administré. Après chaque item administré, le ratio de prédictions correctes est calculé. Si le modèle estime que la probabilité d'un item est de 0,5 ou plus, alors on prédit un succès à cet item, et inversement si la probabilité est plus petite que 0,5. La performance en termes de ratio de bonnes prédictions est calculée de 0 item administré jusqu'à ce que tous les items du test le soient. On présume que les « prédictions » aux items déjà administrés sont correctes, de sorte que la performance culmine nécessairement à 100% de bonnes prédictions.

Le choix du prochain item est basé sur une heuristique qui s'est avérée de performance comparable à l'information de Fisher (traditionnellement utilisée dans des tests adaptatifs) et à la technique basée sur la valeur attendue de la réduction d'entropie (Meshkinfam, 2005). Il s'agit de choisir l'item dont le produit de l'entropie par le nombre de liens est maximal. Cette technique a l'avantage d'être très rapide à calculer sans pour autant résulter en une perte de performance. La formule précise du choix est celle-ci :

$$\max_i E(X_i) \frac{\log(links(X_i + 1) + E(0.5))}{E(0.5)} \quad (8)$$

Pour éviter de créer un biais positif dans la performance, il est nécessaire de ne pas utiliser les mêmes données pour l'entraînement et pour la validation. Environ la moitié des données servent donc à l'entraînement et le l'autre sert pour la simulation (voir la section suivante qui fournit des détails sur les différentes banques de données utilisées).

Comme le processus d'échantillonnage est stochastique, il est aussi nécessaire de procéder à plusieurs simulations basées sur des échantillons différents, ce que l'on nomme en anglais *n-fold validation*. Ce processus permet d'obtenir une valeur plus fiable et évite le biais d'échantillonnage en ne simulant qu'avec un seul échantillon par condition. Ainsi, un total de 10 échantillons (à la fois d'entraînement et de tests) est créé aléatoirement selon la méthode décrite dans la section qui précède. La performance rapportée est basée sur la moyenne et l'écart type de ces 10 simulations.

Une définition plus précise de l'algorithme de simulation est fournie dans la prochaine section.

#### 5.4.1 Algorithme de simulation

Une définition plus précise de l'algorithme de simulation est fournie dans cette section.

Étant donné un test  $\Psi$  et une fonction d'échantillonnage  $sample(\Psi, \frac{|\Psi|}{2})$  qui retourne un échantillon de grosseur  $n = \frac{1}{2}|\Psi|$  en conformité avec à l'algorithme décrit dans la section 5.3.1, alors une simulation consiste en une première boucle dont le nombre d'itérations est le nombre d'expériences voulues (*n-fold*), 10 dans notre étude, une seconde boucle imbriquée où la simulation est effectuée pour chaque sujet de l'expérience, et une troisième boucle imbriquée où chaque item du questionnaire est administré et où l'on compare les prédictions aux items non administrés aux résultats réels. Cet algorithme est le suivant :

$\Psi \leftarrow$  un questionnaire

$n \leftarrow 10$  (nombre d'expériences = 10)

$r3 \leftarrow 0$  performance finale

**itération de 1 à  $n$**

$S_v \leftarrow \text{sample}(\Psi, \frac{|\Psi|}{2})$  (ensemble des sujets dans l'expérience )

$S_t \leftarrow \Psi \setminus S_v$  (ensemble des sujets pour l'entraînement de l'engin d'inférence  $EI$ )

$EI \leftarrow \text{engine}(S_t)$  (initialisation de l'engin d'inférence; création de la structure POKS avec l'échantillon)

**pour chaque  $j$  dans  $S_v$**

$r2 \leftarrow 0$

$Ks_j \leftarrow$  état de la connaissance initial du sujet  $j$

**itération de 1 à  $|\varphi|$**

$r1 \leftarrow 0$

$k \leftarrow$  prochain item à administrer

$o \leftarrow$  succès à l'item  $k$  pour le sujet  $j$

$Ks_j \leftarrow \text{Inference}(EI, Ks_{j-1}, o)$  (état de connaissance du sujet  $j$  calculé avec l'engin d'inférence  $EI$  et basé sur la nouvelle évidence  $o$  et l'état de connaissance précédent)

$r \leftarrow Cl(Ks_j)$  (calcul du taux de succès pour les items non administrés)

$r1 \leftarrow r1 + r$

**fin de la boucle**

$r2 \leftarrow r2 + \frac{r1}{|\varphi|}$

**fin de la boucle**

$r3 \leftarrow r3 + \frac{r2}{|S|}$

**fin de la boucle**

**retourne**( $\frac{r3}{n}$ )

Dans cet algorithme,  $r$  correspond au nombre d'items correctement prédits et classifiés en fonction des items réellement réussis par le sujet  $j$  et tel que calculé par  $Cl(Ks_j)$ . Si la probabilité d'un item,  $P(X_i)$  est au-dessus de 0,5, alors il est estimé comme maîtrisé, autrement il est non maîtrisé. Un item est correctement classifié si la

prédiction correspond à la réalité. L'estimation initiale de l'état de connaissance est basée sur le taux de succès mesuré dans l'échantillon  $S_t$ .

L'algorithme retourne la performance moyenne de classification des prédictions et correspond au calcul suivant :

$$Cl = \frac{\sum_i^r \sum_j^m \sum_k^n M_{ijk}}{rmn}$$

où  $r$  est le nombre total d'items dans le test,  $n$  est le nombre d'expériences (10), et  $m$  est le nombre de sujets dans  $S_v$ .  $M_{ijk}$  représente la classification de la prédiction de l'item  $i$  pour le sujet  $j$  et pour l'expérience  $k$ . Si une prédiction est correcte, la valeur de  $M_{ijk}$  est de 1 et elle est de 0 autrement.

#### 5.4.2 Algorithme d'engine

La fonction Engine( $S$ ) initialise les valeurs  $LN_{i \rightarrow j}$  et  $LS_{i \rightarrow j}$  et établit si une relation  $i \rightarrow j$  existe.

$p_c \leftarrow 0,5$  {seuil de probabilité  $P_c$ }

$\alpha_p \leftarrow 0,5$  {seuil d'erreur pour  $P_c$ }

$\alpha_c \leftarrow 0,5$  {seuil d'erreur pour le test  $X^2$ }

$D[s][q] \leftarrow S$  {  $S$  est la matrice des données de l'échantillon;  $s$  est le nombre de sujets et  $q$  le nombre de questions; les valeurs sont 1 si succès et sinon 0 }

$R[q][q]$  {matrice des liens ( $q$  est le nombre d'items)}

$LN[q][q]$  {matrice symétrique des valeurs de  $LN_{i \rightarrow k}$ }

$LS[q][q]$  {matrice symétrique des valeurs de  $LS_{i \rightarrow k}$ }

```

for all  $i \in S$  do
  for all  $j \in S$  do
    if  $i \neq j$  then
       $a_j \leftarrow 0$  {fréquence de succès pour  $X_i$  étant donné  $X_j$ }
      for all  $k$  in  $D[j][\ ] = 1$  do
         $a_j \leftarrow a_j + D[j][k]$ 
      end for
       $a \leftarrow 0$  {fréquence de succès pour  $X_i$ }
      for all  $k$  in  $D[j][\ ]$  do
         $a \leftarrow a + D[j][k]$ 
      end for
       $t_1 \leftarrow Bp(a_j, a, p_c)$  {test de la binomiale pour  $P(X_j | X_i)$ }
       $t_2 \leftarrow Bp(a_{2j}, a_2, p_c)$  {test de la binomiale pour  $P(X_i | \neg X_i)$ }
       $t_3 \leftarrow X^2(D)$ 
      if  $t_1 = \text{true} \ \& \ t_2 = \text{true} \ \& \ t_3 = \text{true}$  then
         $R[i][j] \leftarrow 1$ 
         $LN[i][j] \leftarrow Odds(a_j / a) / Odds(a / q)$  {la fonction
          Odds(p) retourne la valeur  $p / (1 - p)$ }
         $LS[i][j] \leftarrow Odds(a_{2j} / a) / (1 - Odds(a / q))$ 
      else
         $R[i][j] \leftarrow 0$ 
         $LN[i][j] \leftarrow 1$ 
         $LS[i][j] \leftarrow 1$ 
      end if
    end if
  end for
end for
return  $\{R, LN, LS\}$ 

```

### 5.4.3 Algorithme d'inference

Définition de l'algorithme de la fonction  $Inference(Ie, K_{S_j-1}, o_i)$ .

L'algorithme nécessite :

$Ie$ : valeurs de  $LS_{i \rightarrow k}$  et  $LN_{i \rightarrow k}$  pour tout  $i$  connecté à l'item  $k$ . Si  $o_i$  est vrai (succès). De plus, posons  $Ie(o_i, k) = LS_{i \rightarrow k}$  si  $o_i$  est faux (insuccès) et  $Ie(o_i, k) = LN_{i \rightarrow k}$  sinon.

$K_{S_j-1}$  : l'ensemble des valeurs de chances de chaque item pour le sujet  $j$  avant

l'observation de l'item  $o_i$ . De plus, posons une fonction  $K_{S_j-1}(k)$  qui retourne les chances de succès du sujet  $j$  à l'item  $k$  avant l'observation de l'item  $i$ .

Avec ces éléments, définissons l'algorithme d'inférence qui retourne la valeur de  $K_{S_j}$  : l'état de connaissance du sujet  $j$  après l'observation de l'item  $i$ .

En supposant une observation  $i = o_i$ , l'algorithme met à jour les chances de succès pour les autres items connectés à  $i$  :

$$K_{S_j} = K_{S_j-1}$$

$S_i \leftarrow$  sous-ensemble des items connectés à  $i$  et n'ayant pas encore été observé  
**for all**  $i \in S_j$  **do**

$C_{k0} \leftarrow K_{S_j-1}[k]$  {les chances de succès de  $j$  à l'item  $k$  dans l'état de connaissance avant l'observation de  $o_i$ }

$L_{i \rightarrow k} \leftarrow Ie(o_i, k)$  {le "likelihood of necessity" ou le "likelihood of sufficiency" selon si  $o_i$  est vrai ou faux}

$$C_{k1} \leftarrow C_{k0} \cdot L$$

$$K_{S_j} = \text{Update}(K_{S_{j-1}}, C_{k1})$$

**end for**

**return**  $K_{S_j}$

La fonction  $\text{Update}(K_{S_{j-1}}, C_{k1})$  ne fait que remplacer la valeur des chances de succès de  $k$  par la nouvelle valeur  $C_{k1}$ .

## 5.5 Banque de données

Quatre banques de données ont été utilisées pour les simulations. Elles portent chacune sur un domaine différent et comportent un nombre de questions et de données variables ce qui permet de couvrir différentes conditions. Comme on le verra, cette variabilité a permis de détecter un facteur clé qui expliquera les résultats. Ces quatre banques sont :

1. UNIX : 34 questions répondues par 48 utilisateurs sur les commandes du shell de UNIX. La difficulté de ces questions varie considérablement et va des commandes de base aux commandes plus complexes de traitement de données et de maintenance.
2. ARITHMETIQUE : 20 questions répondues par 149 utilisateurs sur les calculs arithmétiques de fractions. Ces données proviennent de Vomlel (2004) et ont été recueillies auprès d'enfants de 10 à 12 ans.
3. MATHÉMATIQUES : 60 questions de niveau collégial répondues par 426 utilisateurs. Les questions portent principalement sur l'algèbre, le calcul différentiel et la géométrie.
4. FRANÇAIS : 160 questions répondues par 42 utilisateurs sur la compréhension et la lecture en français. Les questions portent autant sur la pragmatique du langage que sur la syntaxe et l'orthographe.

## CHAPITRE 6 RÉSULTATS

Les résultats des simulations sont rapportés aux figures 17 à 20. Elles comportent quatre graphiques de performance, soit un pour chaque banque de données. Les courbes pour les quatre méthodes suivantes sont affichées :

1. **Complètes** : données complètes. Cette situation est fournie à titre comparatif et représente le maximum de performance que l'on peut espérer atteindre.
2. **Uniforme** : échantillonnage aléatoire uniforme des items. Chaque item a la même probabilité d'être sélectionné dans l'échantillon partiel de données.
3. **Plus incertains** : échantillonnage plus fréquent des items les plus incertains. Cette condition correspond à l'utilisation de la fonction  $f_{v1}(X)$  (voir section 5.3.1, page 40).
4. **Moins incertains** : échantillonnage plus fréquents des items les moins incertains. Cette condition correspond à l'utilisation de la fonction  $f_{v2}(X)$  (voir section 5.3.1, page 40).

Pour chaque ensemble de données, le résultat moyen à dix expériences est rapporté, une figure par ensemble de données.

Un intervalle de confiance de 90% est affiché autour de chaque courbe correspondant à la variabilité des 10 expériences par condition.



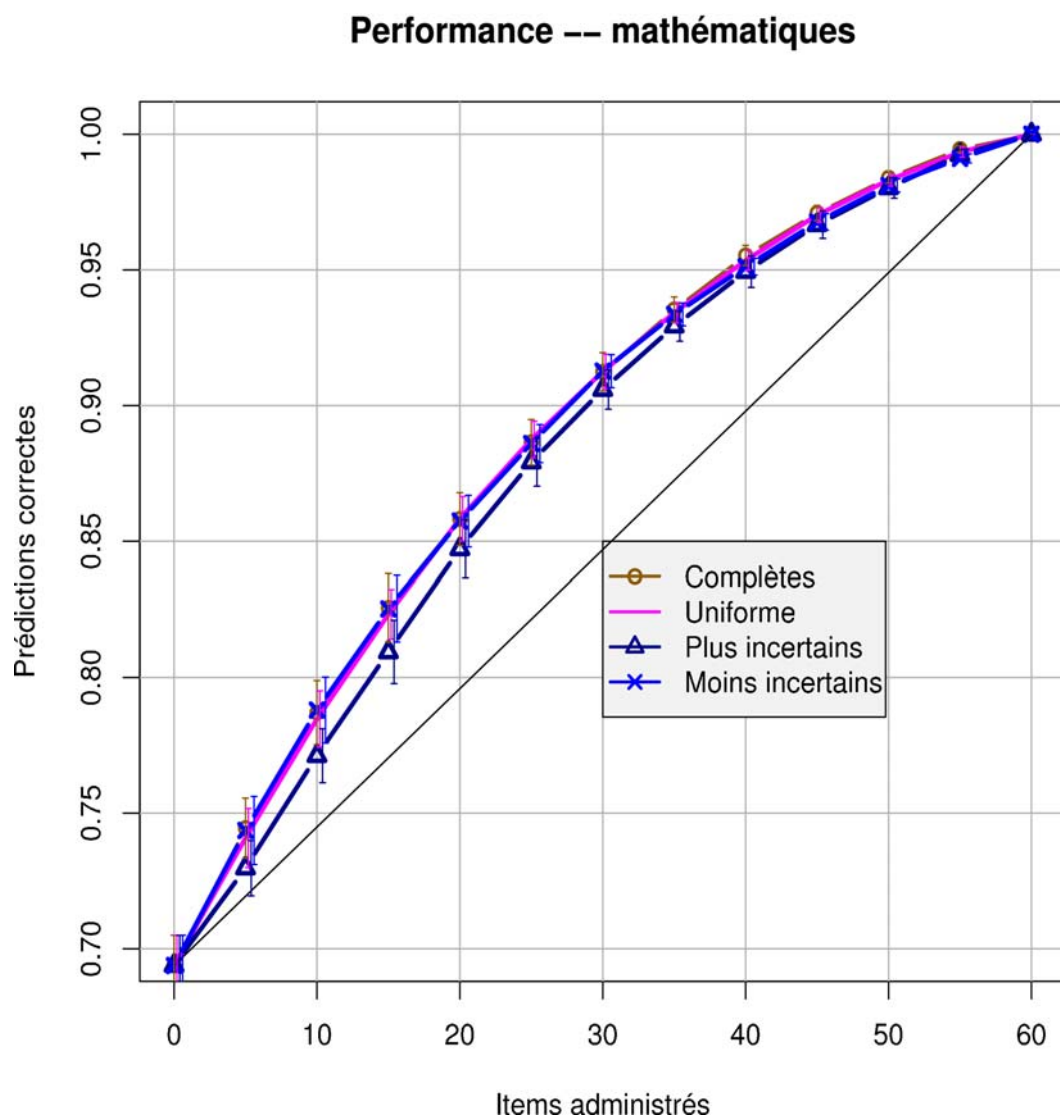


Figure 17 Graphique de performance pour la banque de données Mathématiques

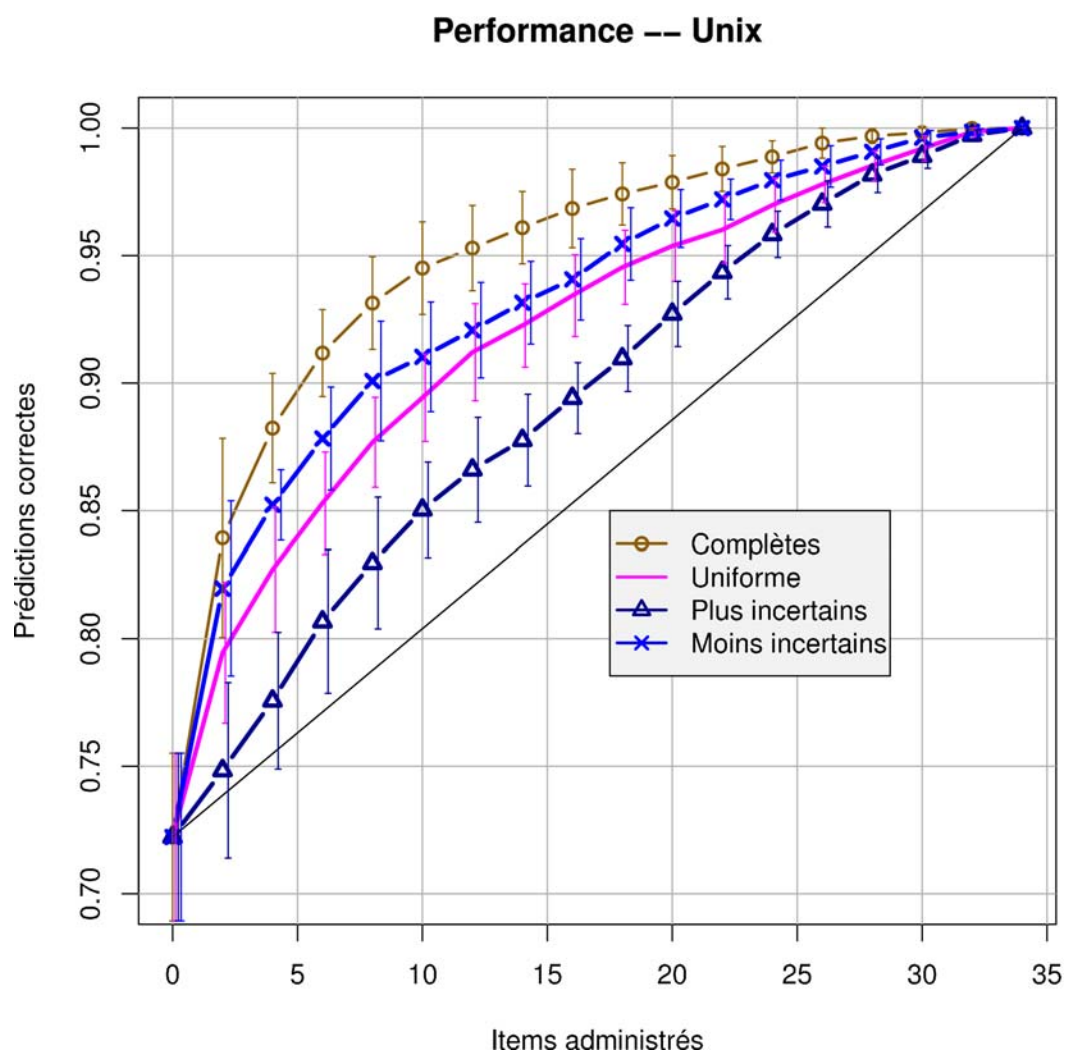


Figure 18 Graphique de performance pour la banque de données Unix

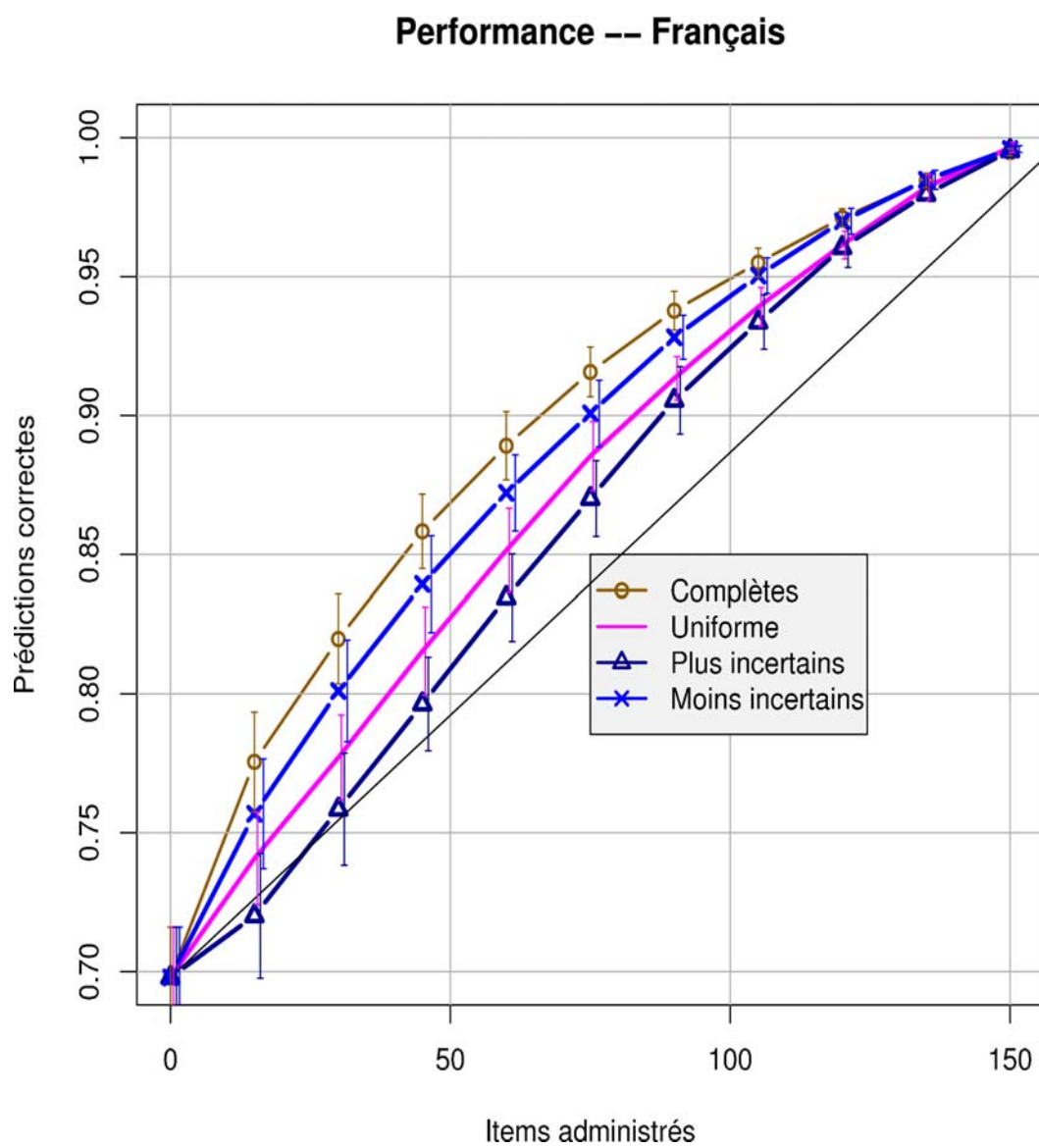


Figure 19 Graphique de performance pour la banque de données Français

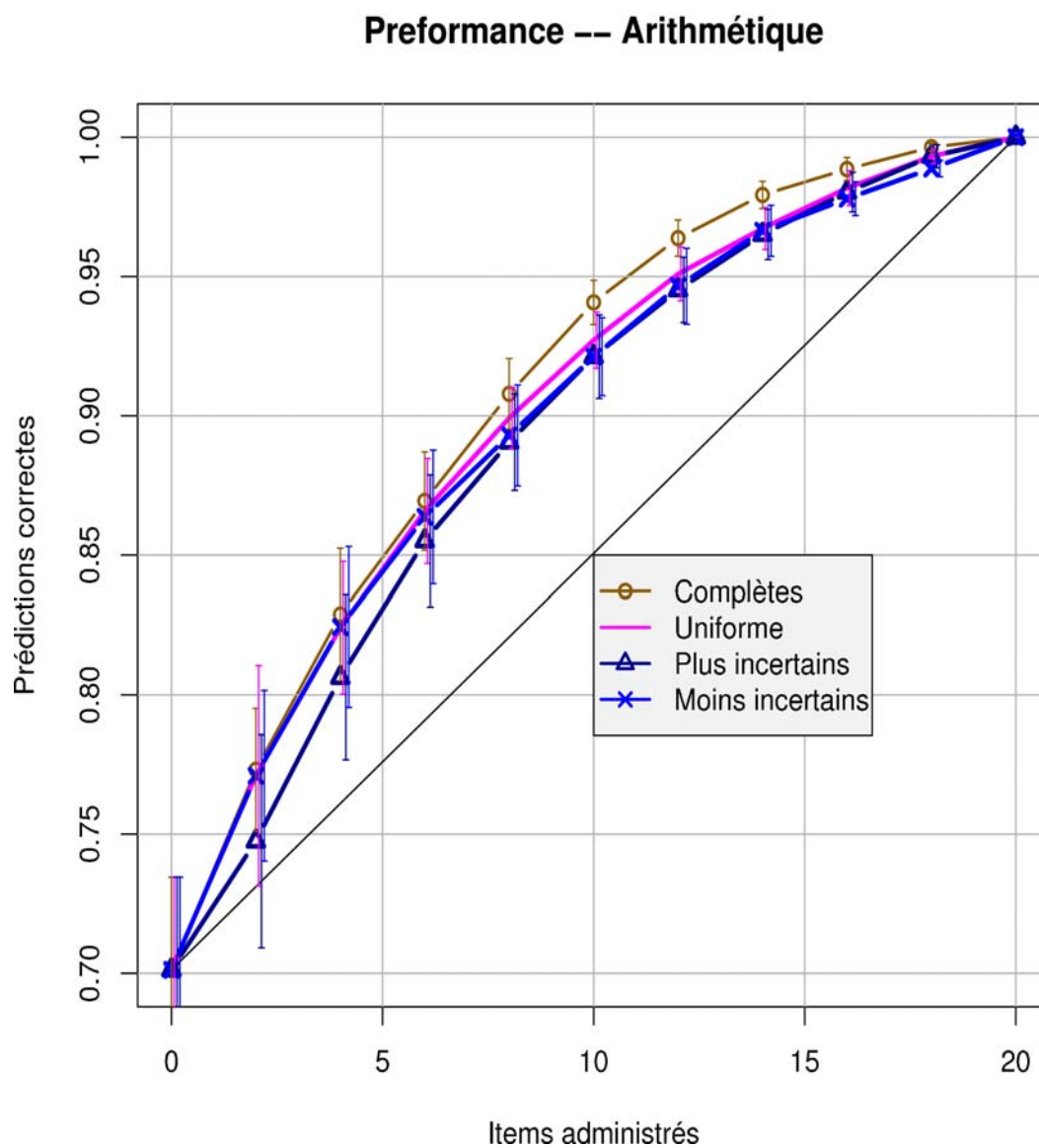


Figure 20 Graphique de performance pour la banque de données Arithmétique

On constate que l'effet des quatre méthodes d'échantillonnage varie en intensité, mais que les résultats sont relativement constants avec la méthode *plus incertains* étant systématiquement la plus basse tandis que la méthode *moins incertain* se situe entre la méthode *uniforme* et l'utilisation des données complètes. Fait à noter, la méthode *plus incertains* est même moins performante que la méthode aléatoire uniforme.

Ces résultats ne concordent pas avec l'hypothèse que nous faisons initialement. Nous avons émis l'hypothèse que les items les plus incertains, donc les items de difficulté moyenne, se retrouveraient à être les plus reliés à d'autres items (puisque'ils se situent dans la couche du milieu du graphique) et que de ce fait ils apportent plus d'information que les autres à la racine ou aux nœuds terminaux du réseau POKS. De plus ces items étant les plus incertains, ils apportent plus d'information et ils ont ainsi tendance à être les premiers choisis lors de la simulation. Nous supposons donc qu'une meilleure précision des probabilités conditionnelles qui sont associées à ces items (provenant d'un échantillonnage qui les favorise), apporterait une plus grande précision au diagnostic et par conséquent une meilleure performance au modèle. Or, même s'il demeure plausible de croire que ces items sont plus reliés que les autres et qu'ils apportent plus d'information en général, un échantillonnage plus fréquent de ces items aux dépens des moins incertains amène au contraire les pires performances des modèles POKS. Au contraire, ce sont les items les moins incertains qui apportent les meilleures performances.

Une piste d'explication se trouve peut-être du côté de la variabilité des effets entre les banques.

À regarder les données de plus près, on remarque que malgré la constance de l'ordre de la performance, c'est la banque Unix qui affiche le plus grand écart entre les conditions, tandis que la banque Mathématique n'affiche que très peu d'écart entre ces mêmes conditions. En fait, on remarque que plus le nombre de sujets est faible, plus l'écart est important entre les méthodes. En effet, l'ordre d'importance des écarts correspond exactement à l'ordre de grandeur du nombre de sujets dans les banques.

Cette observation nous amène à conclure que le bruit d'échantillonnage des petites banques de données semble avoir un effet important, qui est moins présent pour de grandes banques. Le bruit d'échantillonnage pourrait donc être à la source des différences d'effets observées. En poursuivant cette explication, on conclut qu'il est possible que le bruit d'échantillonnage soit plus important pour les items de faible entropie que celles de

forte entropie. En effet, une erreur d'échantillonnage de 1/20 pour estimer une probabilité conditionnelle aux environs de 1/20 est très importante en termes relatifs, mais elle l'est moins pour une probabilité de 10/20. Par exemple, avec un échantillon de 20 observations et une probabilité de 1/20, une erreur de plus ou moins 1 correspond, en termes relatifs, à ce ratio :

$$\frac{\frac{(1 \pm 1) + 1}{20 + 2}}{\frac{1 + 1}{20 + 2}} = \frac{2}{2} = 100\% \quad (9)$$

tandis que pour une probabilité de 10/20, l'erreur relative est plutôt de l'ordre de :

$$\frac{\frac{(10 \pm 1) + 1}{20 + 2}}{\frac{10 + 1}{20 + 2}} = \frac{2}{11} = 18\% \quad (10)$$

Il est donc très plausible qu'il soit plus important d'échantillonner en priorité des items à faible entropie pour éviter de dériver des probabilités conditionnelles fortement erronées qui se retrouveront dans le calcul des inférences (voir chapitre 3) et dont l'impact sera négatif et significatif pour le diagnostic.

Cette explication nécessiterait une nouvelle étude pour la valider. Elle demeure néanmoins une piste très plausible pour expliquer les résultats.

## CONCLUSION

### 7.1 Synthèse des travaux

Nous avons exposé, dans ce mémoire, différentes approches pour effectuer des tests adaptatifs et fait un survol de certaines applications innovatrices qui peuvent en bénéficier. Le problème qui se pose fréquemment en pratique pour ces applications est qu'il faut construire un modèle de diagnostic de la connaissance comportant un grand nombre de questions. Pour un guide d'étude, par exemple, il faut que le diagnostic soit précis pour différents niveaux de connaissances et il faut aussi avoir une banque de questions suffisamment grande pour pouvoir administrer plusieurs quiz avec des questions différentes. Il est donc normal que le nombre de questions soit de plusieurs centaines et il devient alors impossible de demander à un individu de répondre à toutes les questions. Il faut plutôt créer de plus petits questionnaires que l'on donnera à plusieurs individus afin de calibrer un modèle. C'est le problème de l'entraînement avec des données partielles, mais où nous avons la possibilité de déterminer quelles données seront manquantes dans l'échantillon d'entraînement.

Dans ce mémoire, nous avons tenté de déterminer si on peut définir une façon plus efficace de créer des questionnaires et qui est meilleure que celle qui consiste à créer des questionnaires par un choix totalement aléatoire de questions. En d'autres termes, nous avons étudié s'il est possible de définir une méthode d'échantillonnage qui permet d'optimiser la performance d'un modèle donné, en l'occurrence le modèle POKS.

Une méthode simple et applicable consiste à choisir les items selon leur niveau de difficulté ou leur niveau d'incertitude. Il est possible, avec un petit échantillon, d'établir si une question est de forte incertitude (difficulté aux environs de 50% de taux de succès) ou de faible incertitude (difficulté faible, succès près de 100%, ou difficulté forte, succès près de 0%). Notre intuition de départ était que les items à forte incertitude seraient ceux

qui bénéficieraient le plus à être échantillonnées en plus grand nombre, car ils apportent généralement plus d'information. Or, c'est l'inverse qui semble se vérifier.

En effet, les résultats des simulations ont démontré qu'en favorisant l'échantillonnage des items incertains, la performance du modèle à prédire le résultat aux items non répondus était encore plus faible qu'avec un échantillonnage complètement aléatoire. Au contraire, l'échantillonnage favorisant les items les moins incertains produit des modèles significativement plus performants que l'échantillonnage aléatoire et sa performance s'approche même de la performance avec les données complètes.

Une explication à ces résultats pourrait être le bruit qui affecte l'estimation des probabilités conditionnelles pour le modèle POKS. En effet, une erreur de plus ou moins 1 lorsqu'on estime un ratio a beaucoup plus d'impact si la valeur de ce ratio est près de 0 ou de 1, que s'il est près de 0,5. Ces erreurs d'estimation se propagent durant les inférences et créent des biais plus importants pour les probabilités conditionnelles près de 0 et 1, que les erreurs de probabilités conditionnelles aux environs de 0,5. Or, ce sont les items les moins incertains qui se retrouvent le plus souvent dans l'estimation des probabilités conditionnelles aux environs de 0 ou 1.

Cette explication est corroborée par le fait qu'une très faible différence est observable pour les simulations où la calibration est basée sur des banques de données assez grandes, comme c'est le cas pour le test de mathématique et d'arithmétique, alors qu'elle est beaucoup plus marquée pour les simulations basées sur des petites banques comme UNIX et le français. En fait, l'ampleur de la différence suit exactement l'inverse de la taille des banques de calibration.

## **7.2 Limitations des travaux**

De plus, on peut se demander si ces conclusions seraient les mêmes dans le cas du modèle de la TRI, qui est très fréquemment utilisé dans le domaine des tests adaptatifs. Dans ce modèle, on n'estime pas directement les probabilités conditionnelles, mais on essaie plutôt de calibrer une courbe (la courbe de la Figure 12) propre à chaque item en



fonction des données. Il est plausible que l'estimation de cette courbe soit moins fiable pour les items de faible incertitude (difficiles et faciles) et que l'on observe le même phénomène que dans notre étude.

Une des limites de l'approche d'échantillonnage est qu'elle nécessite des statistiques sur les items que l'on veut échantillonner avant même d'avoir pu effectuer cet échantillon. Même si l'estimation d'un taux de succès peut se faire avec un échantillon réduit et qu'il est réaliste de penser qu'un échantillonnage uniforme initial permette de guider correctement la suite de l'échantillonnage basé sur l'entropie (qui dérive du taux de succès), l'étude que nous avons faite ne tient pas compte de ce facteur et les données utilisées étaient celles de l'ensemble des données disponibles. Il faudrait donc vérifier si un échantillon plus réduit atténue les différences que nous avons observées dans les résultats.

### **7.3 Recherches futures**

Si l'hypothèse pour expliquer les résultats obtenus est plausible, nous n'avons toutefois pas validé cette hypothèse avec d'autres simulations. Une telle étude reste à faire dans des travaux futurs.

De même, des expériences seraient à réaliser pour valider si les résultats sont les mêmes pour des modèles de connaissance alternatifs comme la TRI et les réseaux bayésiens.

De plus, il serait intéressant d'explorer d'autres approches d'échantillonnage. Dans cette étude, nous avons choisi une courbe basée sur notre connaissance du domaine (Figure 16), mais elle n'est pas nécessairement la meilleure. Comment déterminer la meilleure courbe à choisir demeure une question ouverte et intéressante à étudier ?

Si les résultats de ces expériences futures démontrent qu'une approche d'échantillonnage basé sur l'entropie des nœuds/items d'un modèle, ou sur une autre propriété, permet d'améliorer les prédictions d'un modèle probabiliste, l'impact serait alors non négligeable. Car les modèles probabilistes sont maintenant très répandus et

plusieurs contextes se prêtent à un échantillonnage contrôlé comme c'est le cas pour les tests adaptatifs. On peut penser à des études cliniques où l'on tente de prédire l'occurrence de maladies à partir de tests médicaux qui sont parfois coûteux à effectuer, mais dont on contrôle souvent l'échantillonnage. Il serait alors possible de diminuer le nombre de tests pour obtenir des modèles dont la performance prédictive est équivalente. Cette piste de recherche s'avère donc prometteuse avec les résultats préliminaires obtenus.

## RÉFÉRENCES

ALEKS CORPORATION. (2008). *ALEKS*. Aleks corporation. Consulté le 12 décembre 2008, Tiré de <http://www.aleks.com>

CONATI, C., GERTNER, A., et VANLEHN, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4), 371-417.

BAKER, F. B., et KIM, S. H. (2004). *Item response theory: parameter estimation techniques*. (2 éd.). New York: Frank B. Baker, Seock-Ho Kim.

BAKER, F. (2001). *Item Response Theory*. EdRes.org. Consulté le 12 décembre 2008, tiré de <http://echo.edres.org:8080/irt/>

BLOOM, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4-16.

BRAINBENCH. *Products*. Brainbench. Consulté le 12 décembre 2008, Tiré de <http://www.brainbench.com/xml/bb/business/products.xml>

DESMARAIS, M. C., MALUF, A., et LIU, J. (1995) User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction*, 5(3-4), 283-315.

DESMARAIS, M. C., et GAGNON, M. (2006). Bayesian Student Models Based on Item to Item Knowledge Structures. *Lecture notes in computer science*, 4227(1), 111-124.

DESMARAIS, M. C., et PU, X. (2005). A Bayesian student model without hidden nodes and its comparison with item response theory. *International Journal of Artificial Intelligence in Education*, 15(4), 291-323.

DESMARAIS, M. C., VILLARREAL, A., et GAGNON, M. (2008). Adaptive Test Design with a Naive Bayes Framework. *Proceedings of the 1<sup>st</sup> Conference on Educational Data Mining 2008, Montréal, Canada* (pp. 48-56). Montréal, Canada: Baker, Barnes, & Beck.

FALMAGNE, J.-C., et DOIGNON, J.-P. (1998). *Knowledge Spaces* (1 éd.). Berlin Heidelberg: Springer-Verlag.

GARRIDO, C. (2006). Different approaches to computer adaptive testing applications.

Consulté le 12 décembre 2008, tire de

<http://cs.joensuu.fi/pages/whamalai/sciwri/cristina.pdf>

HELLER, J., STEINER, S., HOCKEMEYER, C., et ALBERT, D. (2006). Competence-based knowledge structures for personalised learning. *International Journal on E-Learning*, 5(1), 75-88.

LEACTIVEMATH. *System Architecture*. LeActiveMath. Consulté le 12 décembre 2008,

Tiré de <http://www.leactivemath.org/system-architecture.html>

MELIS, J., ET SIEKMANN, J. (2004) ActiveMath : An intelligent tutoring system for mathematics. *Actes du Seventh International Conference on Artificial Intelligence and Soft Computing (ICAISC2004)*, LNCS 3070, Springer-Verlag. pp. 91-101.

MARGINEANTU, D. D., et DIETTERICH, T. G. (2002) Improved class probability estimates from decision tree models. in D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, and B. Yu (Eds.) *Nonlinear Estimation and Classification; Lecture Notes in Statistics*, 171, pp. 169-184. New York: Springer-Verlag.

MARTINEZ, M. (2002). What is personalized learning. *The eLearning developers'journal*, publié en ligne le 7 mai 2002, 1-8. Consulté le 12 décembre 2008, tiré de <http://www.elearningguild.com/pdf/2/050702dss-h.pdf>

MESHKINFAM, P. (2005). *Bayesian modeling with strong vs. weak assumptions in the domain of skills assessment*. M.Sc.A. École Polytechnique de Montréal, Québec, Canada.

MILLAN, E., et PEREZ-DE-LA-CRUZ, J.-L. (2002). A Bayesian diagnostic Algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction*, 12(2-3), 281-330.

MISLEVY, R. J., et ALMOND, R. G. (1999). Graphical models and computerized adaptive testing. *Applied psychological measurement*, 23(3), 223-237.

RUSSEL, S. J., et NORVIG, P. (2003). Probabilistic Reasoning. *Artificial intelligence a modern approach* (2 éd., pp. 492-533). New Jersey: Prentice Hall Series in Artificial Intelligence.

UNIVERSITY OF PITTSBURGH. *Reasons for using Andes*. University of Pittsburgh.  
Consulté le 12 décembre 2008, Tiré de <http://www.andestutor.org/Pages/Reasons.html>

VOMLEL, J. (2004). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 12(1), 83-100.

WEBER, G., KUHL, H. C., et WEIBELZAHN, S. (2001). Developing Adaptive Internet Based Courses with the Authoring System NetCoach. *Lecture Notes In Computer Science*, 2266, 226-238.

XIAOLIN, Niu., MCCALLA, G., VASSILEVA, J. (2004). Purpose-Based Expert Finding in a Portafolio Management System. *Computational Intelligence*, 20(4), 548-561.